



**COMBINAISON DE CNN ET LSTM POUR LA RECONNAISSANCE
D'ACTIVITÉS DANS UN ENVIRONNEMENT INTELLIGENT AVEC RADARS
ULTRA-WIDEBAND**

PAR ALEXANDRE BEAULIEU

**MÉMOIRE PRÉSENTÉ À L'UNIVERSITÉ DU QUÉBEC À CHICOUTIMI
COMME EXIGENCE PARTIELLE EN VUE DE L'OBTENTION DU GRADE DE
MAÎTRE ÈS SCIENCES EN INFORMATIQUE**

QUÉBEC, CANADA

© ALEXANDRE BEAULIEU, 2021

RÉSUMÉ

Dans les dernières années, la population mondiale a souffert d'un vieillissement. Cette tendance vise à croître dans les prochaines années. Un des problèmes qui en découle est que le nombre de personnes nécessitant du soutien au quotidien devrait également croître avec le vieillissement de la population. Les habitats intelligents offrent une alternative intéressante pouvant soulager une partie de ce problème. Les habitats intelligents contiennent une panoplie de capteurs qui, combinés à l'intelligence artificielle, ont le potentiel de supporter leurs occupants dans leur quotidien. Au centre de cette solution technologique se trouve la reconnaissance d'activités. La reconnaissance d'activités est un problème complexe visant à identifier quelle activité est effectuée par une personne. La complexité de ce problème découle de la nature des activités à identifier, des capteurs disponibles et de l'impact de ces derniers sur la vie privée. Parmi les capteurs exploitables pour la reconnaissance d'activités, les radars *Ultra-Wideband* (UWB) ont généré beaucoup d'intérêt dans les dernières années, car ils offrent un compromis intéressant entre précision et protection de la vie privée. Les travaux couverts par ce mémoire visent à atteindre un meilleur taux de reconnaissance d'activités de la vie quotidienne que les travaux précédents en combinant réseaux de neurones et radar UWB.

Pour ce faire, deux approches ont été testées. Dans un premier temps, la combinaison de système experts et de réseau de neurones est explorée. Ensuite, un nouveau modèle d'apprentissage profond combinant différentes architectures (*Long Short-Term Memory* (LSTM) et réseau de neurones convolutif (CNN)) est développé pour faire la reconnaissance d'activités. Ce modèle a comme avantage de fusionner des données provenant de différentes sources et exploite des données prétraitées minimalement. Ce modèle, appelé *Tuned EfficientNetB0 avec LSTM*, offre une amélioration de 18,63 % du taux de reconnaissance par rapport aux réseaux de neurones développés dans des travaux antérieurs exploitant le même jeu de données. Finalement, un nouveau jeu de données a été construit et utilisé pour identifier la source des limitations de la reconnaissance d'activité dans le contexte actuel.

ABSTRACT

In the last few years, the world population has gotten older. This trend is expected to continue in the future. One of the problems related to this phenomenon is the increasing number of persons requiring assistance in their everyday life. The number of person requiring assistance is expected to rise along with the aging population. Smart homes are becoming a more and more compelling alternative to direct human supervision. Smart homes are equipped with sensors that, coupled with artificial intelligence, can support their occupants whenever needed. At the heart of this technological solution lies the problem of activity recognition. Activity Recognition is a complex problem, due to the variable nature of activities recognized, the variety of sensors available and their respective impact on privacy. Among the variety of sensors exploitable for activity recognition, Ultra-Wideband (UWB) radars have raised a lot of interest due to their precision and lower impact on privacy. Hence, the work presented in this thesis aims at acheiving higher accuracy than previous works for activity of daily living using UWB and neural networks.

To do so, two approaches were investigated. Firstly, a solution combining a neural network with an Expert System was developed. The second approach consisted in the development of a new neural network combining Long Short-Term Memory (LSTM) layers and Convolutionnal Neural Network (CNN) layers for the task of activity recognition. This model has the advantage of fusing data sources and requires only a minimalist preprocessing of the UWB radar data. This model, called *Tuned EfficientNetB0 with LSTM*, showed an improvement of 18.63 % in accuracy over a neural network developed in previous works on the same dataset. Finally, a new dataset was created to investigate the cause of limitations in the recognition of some activities.

TABLE DES MATIÈRES

LISTE DES TABLEAUX	vi
LISTE DES FIGURES	vii
LISTE DES ABRÉVIATIONS	xii
CHAPITRE I – INTRODUCTION	1
1.1 CONTEXTE DE LA RECHERCHE	1
1.1.1 VIEILLISSEMENT DE LA POPULATION	1
1.1.2 ACTIVITÉS DE LA VIE QUOTIDIENNE	2
1.1.3 ENVIRONNEMENTS INTELLIGENTS	3
1.1.4 RECONNAISSANCE D’ACTIVITÉS	5
1.2 PROBLÉMATIQUE DE RECHERCHE	9
1.3 CONTRIBUTION DE LA RECHERCHE	11
1.4 MÉTHODOLOGIE DE LA RECHERCHE	12
1.5 ORGANISATION DU MÉMOIRE	14
CHAPITRE II – REVUE DE LA LITTÉRATURE	16
2.1 DÉFINITIONS	16
2.1.1 RADAR ULTRA-WIDEBAND	16
2.1.2 APPRENTISSAGE PROFOND	23
2.1.3 SYSTÈMES EXPERTS	43
2.2 CAS APPLIQUÉS	44
2.2.1 RECONNAISSANCE D’AVQ	44
2.2.2 APPLICATION DES RADARS ULTRA-WIDEBAND.	51
2.2.3 TRAVAUX ANTÉRIEURS	61
CHAPITRE III – CRÉATION D’UN MODÈLE DE RECONNAISSANCE D’ACTIVITÉS	64
3.1 INTRODUCTION	64

3.2	JEU DE DONNÉES	64
3.3	EXPLORATION DE L'UTILISATION D'UN SYSTÈMES EXPERTS	70
3.3.1	MODÈLE DÉVELOPPÉ DANS LES TRAVAUX ANTÉRIEURS	71
3.3.2	SYSTÈME EXPERT	72
3.3.3	RÉSULTATS ET DISCUSSION	79
3.4	DÉVELOPPEMENT D'UN MODÈLE D'APPRENTISSAGE PROFOND	85
3.4.1	TRAITEMENT DE DONNÉES	86
3.4.2	ARCHITECTURE PROPOSÉE	88
3.4.3	RÉSULTATS ET DISCUSSION	94
3.5	ANALYSE DE LA CAPACITÉ DES RADARS UWB POUR LA CAPTATION DE MOUVEMENTS AU LIARA	100
3.5.1	NOUVEAU JEU DE DONNÉES	100
3.5.2	VISUALISATION DES DONNÉES	103
3.5.3	ANALYSE DE LA PERFORMANCE DU MODÈLE SUR LE NOU- VEAU JEU DE DONNÉES	108
3.5.4	RÉSULTATS ET DISCUSSION	110
	CHAPITRE IV – CONCLUSION	117
4.1	REVUE DES CONTRIBUTIONS	117
4.2	LIMITES	119
4.3	TRAVAUX FUTURS	120
	BIBLIOGRAPHIE	122
	APPENDICE A – MATRICES DE CONFUSION	134
	APPENDICE B – TAUX DE RECONNAISSANCE	136
	APPENDICE C – MATRICES DE CONFUSION SUR LE JEU DE DONNÉES DE MOUVEMENTS	141
	APPENDICE D – CERTIFICAT D'APPROBATION ÉTHIQUE	149

LISTE DES TABLEAUX

TABLEAU 3.1 :	DURÉES DES DIFFÉRENTES ACTIVITÉS.	66
TABLEAU 3.2 :	ARCHITECTURE DE L'AUTOENDEUR DÉVELOPPÉ.	89
TABLEAU 3.3 :	PERFORMANCE DE DIFFÉRENTS MODÈLES SUR LE JEU DE DONNÉES DÉVELOPPÉ À PARTIR DE SCANS DE RADARS UWB.	90
TABLEAU 3.4 :	VARIATIONS DE LA TAILLE DES FILTRES ET LEUR PERFOR- MANCE RESPECTIVES.	92
TABLEAU 3.5 :	ARCHITECTURE DU MODÈLE <i>LSTM</i>	93
TABLEAU 3.6 :	COMPARAISON DE LA PERFORMANCE DES MODÈLES SUR DES SOUS-ENSEMBLES DU JEU DE DONNÉES LOSO 1.	100
TABLEAU 3.7 :	COMPARAISON DU TAUX DE RECONNAISSANCE MOYEN DU MODÈLE <i>TUNED EFFICIENTNETB0</i> SUR L'ENSEMBLE DES JEUX DE DONNÉES LOSO.	111

LISTE DES FIGURES

FIGURE 2.1 – VISUALISATION DE L’EFFET DOPPLER : A) EFFET DOPPLER SUR UN OBJET STATIQUE. B) EFFET DOPPLER SUR UN OBJET S’ÉLOIGNANT DE LA SOURCE. C) EFFET DOPPLER SUR UN OBJET SE RAPPROCHANT DE LA SOURCE.	18
FIGURE 2.2 – UN RETOUR DE DONNÉES REPRÉSENTANT UNE DONNÉE DISTANCE-DOPPLER PAR UN RADAR X4M200 TIRÉ DE NOVELDA AS (2018). L’AXE HORIZONTAL REPRÉSENTE LA DISTANCE (EN MÈTRES) ET L’AXE VERTICAL L’AMPLITUDE EN DB.	22
FIGURE 2.3 – DONNÉES BRUTES ENREGISTRÉES LORS DE L’EXÉCUTION D’UNE ACTIVITÉ PAR UNE PERSONNE DANS UN HABITAT INTELLIGENT PAR 3 RADARS PLACÉS À DES ENDROITS DIFFÉRENTS.. . . .	23
FIGURE 2.4 – STRUCTURE D’UN PERCEPTRON.	25
FIGURE 2.5 – COUCHE CONVOLUTIVE.	33
FIGURE 2.6 – COUCHE DE MISE EN COMMUN.	35
FIGURE 2.7 – ARCHITECTURE D’UN RÉSEAU DE NEURONES CONVOLUTIFS.	36
FIGURE 2.8 – EXEMPLE SIMPLISTE D’UN NEURONE AVEC CYCLE RÉCURSIF.	36
FIGURE 2.9 – NEURONE DE LSTM.	37
FIGURE 3.1 – PLAN DE L’APPARTEMENT INTELLIGENT DU LIARA. LA POSITION DES RADARS UWB EST INDIQUÉE EN ROUGE. CETTE FIGURE EST TIRÉE DE MAITRE ET AL. (2021). REPRODUIT AVEC LA PERMISSION DE ELSEVIER.	65
FIGURE 3.2 – UN SCAN CAPTÉ PAR LE XETHERU X4M200 (DISTANCE-DOPPLER) AU LIARA. FIGURE TIRÉE DE MAITRE ET AL. (2021). REPRODUIT AVEC LA PERMISSION DE ELSEVIER.	68
FIGURE 3.3 – EXEMPLE VISUEL DE LA TRILATÉRATION.	69

FIGURE 3.4 – ARCHITECTURE DU MODÈLE <i>CNN-LSTM</i> . FIGURE TIRÉE DE MAITRE ET AL. (2021). REPRODUIT AVEC LA PERMISSION DE ELSEVIER.	72
FIGURE 3.5 – ARCHITECTURE DU SYSTÈME EXPERT.	74
FIGURE 3.6 – TAUX DE RECONNAISSANCE DU MODÈLE <i>CNN-LSTM</i> SUR LE JEU DE DONNÉES LOSO 1.	77
FIGURE 3.7 – COMPARAISON DU TAUX DE RECONNAISSANCE BALANCÉ DU TOP-4 DU MODÈLE CENTRAL (<i>CNN-LSTM</i>) ET DU SYSTÈME EXPERT SUR L'ENSEMBLE DES CLASSES, LES CLASSES LONGUES ET LES CLASSES COURTES SUR LE JEU DE DONNÉES D'ENTRAÎNEMENT LOSO 1.	81
FIGURE 3.8 – COMPARAISON DU TAUX DE RECONNAISSANCE BALANCÉ DU TOP-4 DU MODÈLE CENTRAL (<i>CNN-LSTM</i>) ET DU SYSTÈME EXPERT SUR L'ENSEMBLE DES CLASSES, LES CLASSES LONGUES ET LES CLASSES COURTES DU JEU DE DONNÉES DE TEST LOSO 1.	82
FIGURE 3.9 – COMPARAISON DU TAUX DE RECONNAISSANCE BALANCÉ DU TOP-4 DU MODÈLE CENTRAL (<i>CNN-LSTM</i>) ET DU SYSTÈME EXPERT SUR L'ENSEMBLE DES CLASSES, LES CLASSES LONGUES ET LES CLASSES COURTES SUR LE JEU DE DONNÉES D'ENTRAÎNEMENT LOSO 2.	83
FIGURE 3.10 – COMPARAISON DU TAUX DE RECONNAISSANCE BALANCÉ DU TOP-4 DU MODÈLE CENTRAL (<i>CNN-LSTM</i>) ET DU SYSTÈME EXPERT SUR L'ENSEMBLE DES CLASSES, LES CLASSES LONGUES ET LES CLASSES COURTES DU JEU DE DONNÉES DE TEST LOSO 2.	84
FIGURE 3.11 – DEUX ACTIVITÉS PRÉTRAITÉES POUR LE MODÈLE D'APPRENTISSAGE PROFOND. A) ACTIVITÉ <i>MARCHER</i> . B) ACTIVITÉ <i>FAIRE LE MÉNAGE</i>	87
FIGURE 3.12 – ARCHITECTURE DU MODÈLE FINAL.	94
FIGURE 3.13 – COMPARAISON DU TAUX DE RECONNAISSANCE MOYEN DES MODÈLES SUR L'ENSEMBLE DES JEUX DE DONNÉES LOSO.	97

FIGURE 3.14 – COMPARAISON DU TAUX DE RECONNAISSANCE BALANCÉ MOYEN DES MODÈLES SUR L'ENSEMBLE DES JEUX DE DONNÉES LOSO.	97
FIGURE 3.15 – COMPARAISON DE L'ÉCART TYPE DU TAUX DE RECONNAISSANCE DES MODÈLES SUR L'ENSEMBLE DES JEUX DE DONNÉES LOSO.	98
FIGURE 3.16 – COMPARAISON DE L'ÉCART TYPE DU TAUX DE RECONNAISSANCE BALANCÉ DES MODÈLES SUR L'ENSEMBLE DES JEUX DE DONNÉES LOSO.	98
FIGURE 3.17 – EMBLEMENTS PRÉDÉTERMINÉS POUR LA CRÉATION DU NOUVEAU JEU DE DONNÉES AU LIARA.. . . .	101
FIGURE 3.18 – DIFFÉRENTS MOUVEMENTS PLANIFIÉS POUR LA COLLECTE DE DONNÉES. A) AUCUN MOUVEMENT. B) MOUVEMENTS FINS. C) MOUVEMENTS INTERMÉDIAIRES. D) MOUVEMENTS GRANDS.	102
FIGURE 3.19 – COMPARAISON DES DONNÉES BRUTES DES MOUVEMENTS ENREGISTRÉS À LA POSITION 1 AVEC CAPTEURS INSTALLÉS À 36 CM DU SOL. A) <i>AUCUN MOUVEMENT</i> B) <i>MOUVEMENTS FINS</i> C) <i>MOUVEMENTS INTERMÉDIAIRES</i> D) <i>MOUVEMENTS GRANDS</i>	104
FIGURE 3.20 – COMPARAISON DES DONNÉES BRUTES DU RADAR UWB 1 POUR L'ENSEMBLE DES MOUVEMENTS À LA POSITION 1 AVEC CAPTEUR INSTALLÉ À 36 CM DU SOL. A) <i>AUCUN MOUVEMENT</i> B) <i>MOUVEMENTS FINS</i> C) <i>MOUVEMENTS INTERMÉDIAIRES</i> D) <i>MOUVEMENTS GRANDS</i>	105
FIGURE 3.21 – COMPARAISON DES DONNÉES BRUTES DE L'ACTIVITÉ <i>MOUVEMENTS GRANDS</i> AUX POSITIONS 2, 4 ET 9 AVEC CAPTEURS INSTALLÉS À 36 CM DU SOL. A) POSITION 2 B) POSITION 4 C) POSITION 9	106
FIGURE 3.22 – COMPARAISON DES DONNÉES TRAITÉES DES ACTIVITÉS À LA POSITION 1 AVEC CAPTEURS INSTALLÉS À 36 CM DU SOL. A) <i>AUCUN MOUVEMENT</i> B) <i>MOUVEMENTS FINS</i> C) <i>MOUVEMENTS INTERMÉDIAIRES</i> D) <i>MOUVEMENTS GRANDS</i>	107

FIGURE 3.23 – COMPARAISON DES DONNÉES TRAITÉES DE L’ACTIVITÉ <i>MOUVEMENTS GRANDS</i> AUX POSITIONS 2, 4 ET 9 AVEC CAPTEURS INSTALLÉS À 36 CM DU SOL. A) POSITION 2 B) POSITION 4 C) POSITION 9.	108
FIGURE 3.24 – COMPARAISONS DES MATRICES DE CONFUSIONS DE <i>TUNED EFFICIENTNETB0</i> SOMMÉES POUR L’ENTIÈRETÉ DES JEUX DE DONNÉES LOSO, POUR CHACUNE DES COMBINAISONS DE HAUTEURS DE CAPTEURS ET DE MÉTHODE D’APPRENTISSAGE. A) APPRENTISSAGE À PARTIR DE ZÉRO AVEC RADARS UWB PLACÉS À 36 CM DU SOL. B) APPRENTISSAGE PAR TRANSFERT AVEC RADARS UWB PLACÉS À 36 CM DU SOL. C) APPRENTISSAGE À PARTIR DE ZÉRO AVEC RADARS UWB PLACÉS À 96 CM DU SOL. D) APPRENTISSAGE PAR TRANSFERT AVEC RADARS UWB PLACÉS À 96 CM DU SOL.	112
FIGURE 3.25 – COMPARAISONS DU TAUX DE RECONNAISSANCE DE <i>TUNED EFFICIENTNETB0</i> POUR CHACUNE DES COMBINAISONS DE HAUTEURS DE CAPTEURS ET DE MÉTHODE D’APPRENTISSAGE EN FONCTION DE LA POSITION OÙ SE DÉROULE LES MOUVEMENTS DE BRAS. LES POSITIONS 9 ET 10 OUVERT CORRESPONDENT AUX LOSO 9 ET 10, ET LES POSITIONS 9 ET 10 FERMÉ CORRESPONDENT AUX LOSO 11 ET 12.	115
FIGURE A.1 – MATRICE DE CONFUSION DU MODÈLE CNN-LSTM SUR LE JEU DE DONNÉES LOSO 1.	134
FIGURE A.2 – MATRICE DE CONFUSION DU MODÈLE <i>TUNED EFFICIENTNETB0</i> AVEC LSTM SUR LE JEU DE DONNÉES LOSO 1.	135
FIGURE B.1 – TAUX DE RECONNAISSANCE DU MODÈLE <i>TUNED EFFICIENTNETB0</i> SUR LE JEU DE DONNÉES DE MOUVEMENTS AVEC RADARS UWB INSTALLÉS À 36 CM DU SOL.	137
FIGURE B.2 – TAUX DE RECONNAISSANCE DU MODÈLE <i>TUNED EFFICIENTNETB0</i> AVEC ENTRAÎNEMENT PAR TRANSFERT SUR LE JEU DE DONNÉES DE MOUVEMENTS AVEC RADARS UWB INSTALLÉS À 36 CM DU SOL.. . . .	138
FIGURE B.3 – TAUX DE RECONNAISSANCE DU MODÈLE <i>TUNED EFFICIENTNETB0</i> SUR LE JEU DE DONNÉES DE MOUVEMENTS AVEC RADARS UWB INSTALLÉS À 96 CM DU SOL.	139

FIGURE B.4 – TAUX DE RECONNAISSANCE DU MODÈLE TUNED EFFICIENT-NETB0 AVEC ENTRAÎNEMENT PAR TRANSFERT SUR LE JEU DE DONNÉES DE MOUVEMENTS AVEC RADARS UWB INSTALLÉS À 96 CM DU SOL.	140
FIGURE C.1 – MATRICES DE CONFUSION DU MODÈLE TUNED EFFICIENT-NETB0 SUR LE JEU DE DONNÉES DE MOUVEMENTS AVEC RADARS UWB INSTALLÉS À 36 CM DU SOL, LOSO 1 À 6.	141
FIGURE C.2 – MATRICES DE CONFUSION DU MODÈLE TUNED EFFICIENT-NETB0 SUR LE JEU DE DONNÉES DE MOUVEMENTS AVEC RADARS UWB INSTALLÉS À 36 CM DU SOL, LOSO 7 À 12.	142
FIGURE C.3 – MATRICES DE CONFUSION DU MODÈLE TUNED EFFICIENT-NETB0 SUR LE JEU DE DONNÉES DE MOUVEMENTS AVEC RADARS UWB INSTALLÉS À 96 CM DU SOL, LOSO 1 À 6.	143
FIGURE C.4 – MATRICES DE CONFUSION DU MODÈLE TUNED EFFICIENT-NETB0 SUR LE JEU DE DONNÉES DE MOUVEMENTS AVEC RADARS UWB INSTALLÉS À 96 CM DU SOL, LOSO 7 À 12.	144
FIGURE C.5 – MATRICES DE CONFUSION DU MODÈLE TUNED EFFICIENT-NETB0 AVEC APPRENTISSAGE PAR TRANSFERT SUR LE JEU DE DONNÉES DE MOUVEMENTS AVEC RADARS UWB INSTALLÉS À 36 CM DU SOL, LOSO 1 À 6.	145
FIGURE C.6 – MATRICES DE CONFUSION DU MODÈLE TUNED EFFICIENT-NETB0 AVEC APPRENTISSAGE PAR TRANSFERT SUR LE JEU DE DONNÉES DE MOUVEMENTS AVEC RADARS UWB INSTALLÉS À 36 CM DU SOL, LOSO 7 À 12.	146
FIGURE C.7 – MATRICES DE CONFUSION DU MODÈLE TUNED EFFICIENT-NETB0 AVEC APPRENTISSAGE PAR TRANSFERT SUR LE JEU DE DONNÉES DE MOUVEMENTS AVEC RADARS UWB INSTALLÉS À 96 CM DU SOL, LOSO 1 À 6.	147
FIGURE C.8 – MATRICES DE CONFUSION DU MODÈLE TUNED EFFICIENT-NETB0 AVEC APPRENTISSAGE PAR TRANSFERT SUR LE JEU DE DONNÉES DE MOUVEMENTS AVEC RADARS UWB INSTALLÉS À 96 CM DU SOL, LOSO 7 À 12.	148

LISTE DES ABRÉVIATIONS

AVQ	Activité de la Vie Quotidienne
CNN	Convolutional Neural Network
DFT	Discrete Fourier Transform
DIM	Département d'Informatique et de Mathématique
DOMUS	Domotique et informatique mobile à l'Université de Sherbrooke
FFT	Fast Fourier Transform
GAN	Generative Adversarial Network
ILSVRC	ImageNet Large Scale Visual Recognition Competition
IoT	Internet of Things
KNN	K-Nearest Neighbors
LIARA	Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités
LOSO	Leave One Subject Out
LSTM	Long Short-Term Memory
MOCAP	Motion Capture
MSE	Mean Squared Error
MSSA	Multivariate Singular Spectral Analysis
PIR	Passive Infrared
ReLU	Rectified Linear Unit
RGB	Red Green Blue
RNN	Recurrent Neural Network
SVM	Support Vector Machine
UQAC	Université du Québec À Chicoutimi
UWB	Ultra-Wideband
VMD	Variational Mode Decomposition

REMERCIEMENTS

Je tiens à remercier M. Kévin Bouchard, professeur au Département d'Informatique et de Mathématique (DIM) de l'Université du Québec À Chicoutimi (UQAC) et membre actif du Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités (LIARA), pour l'opportunité de poursuivre ce projet de recherche sous sa direction.

J'aimerais également remercier M. Julien Maître, professeur au DIM de l'UQAC et membre actif du LIARA, et M. Sébastien Gaboury, professeur au DIM de l'UQAC et directeur du LIARA, pour leurs commentaires et leur support en tant que co-directeur du projet.

Finalement, je voudrais remercier l'entièreté de l'équipe du LIARA pour leur accueil au laboratoire.

CHAPITRE I

INTRODUCTION

1.1 CONTEXTE DE LA RECHERCHE

1.1.1 VIEILLISSEMENT DE LA POPULATION

Selon le "World Population Prospect 2019" du département des affaires économiques et sociales de l'Organisation des Nations Unies (United Nations Department of Economic and Social Affairs , 2019), le nombre de personnes âgées de plus de 65 ans pourrait atteindre 25% de la population en Amérique du Nord d'ici 2050. Globalement, le nombre de personnes de plus de 80 ans devrait tripler dans le même laps de temps.

Avec l'âge vient un lot de maladies et de conditions qui affectent les capacités d'une personne à effectuer les activités de la vie quotidienne, comme le faire ménage, faire à manger, prendre des médicaments et autres tâches reliées à l'hygiène personnelle. Cette perte d'autonomie nuit considérablement à la capacité d'une personne à maintenir son domicile. Dans les dernières années, une visibilité accrue a été apportée à une de ces maladies : la maladie d'Alzheimer. Celle-ci provoque au stade précoce des pertes de mémoire en plus de moments de confusions, et entraîne dans les stades les plus avancés des symptômes de démence sévère et une incapacité importante à interagir avec son environnement (Alzheimer's Association, 2018). Plus généralement, il est estimé que le nombre de personnes atteintes de démence dans le monde pourrait passer de 46,8 millions à 131,5 millions d'ici 2050 (Prince

et al., 2015). Au Canada, le nombre de personnes atteintes de démence passera de 564 000 en 2016 à 937 000 d'ici 2031 (Alzheimer Society of Canada, 2016). La perte de facultés a un impact important non seulement sur les personnes qui en souffrent, mais aussi sur les proches qui prennent soin de ces personnes. Selon un rapport de l'Alzheimer Association sur l'impact de cette maladie aux États-Unis (Alzheimer's Association, 2018), un proche aidant qui prend soin d'une personne atteinte de démence donne en moyenne 21,9 heures de son temps chaque semaine pour offrir du soutien et des soins, et est plus à risque de souffrir de stress, de dépression, et d'éprouver des difficultés émotionnelles et financières substantielles.

Malheureusement, la maladie d'Alzheimer est seulement une petite facette des différents troubles qui peuvent survenir avec le vieillissement normal d'une personne. Le vieillissement est également associé à une diminution des capacités physiques (Kenny *et al.*, 2008) et cognitives (Fillit *et al.*, 2002) d'une personne et à un risque accru de chutes (van Weel *et al.*, 1995), ce qui peut engendrer de lourdes conséquences comme des blessures, mineures ou graves, et une perte d'autonomie.

1.1.2 ACTIVITÉS DE LA VIE QUOTIDIENNE

Une activité de la vie quotidienne, ou AVQ, est une activité qui est effectuée par une personne au quotidien et qui a pour but le maintien de sa propre personne. *Manger, se laver, faire le ménage, s'habiller* sont des exemples notables d'AVQ. Le concept d'AVQ est apparu à la fin des années 1950 grâce entre autres au travail du docteur Sidney Katz (Noelker et Browdie, 2013). La capacité d'une personne à réaliser des AVQ était à l'origine utilisée pour

mesurer l'indépendance d'une personne. Plus la personne peut réaliser d'AVQ, et réaliser des AVQ considérées comme plus complexes, plus la personne est indépendante (peut donc prendre soin d'elle-même).

Lorsqu'une personne n'est plus capable d'effectuer les AVQ seule, deux options s'offrent à elle ; d'un côté elle peut recevoir de l'aide à la maison sous la forme de proche aidant et de l'autre déménager dans des installations où cette aide lui sera offerte (comme les résidences pour personnes âgées). Déménager pour une personne d'un certain âge peut être source de beaucoup d'anxiété (Vasara, 2015). Il est donc souhaitable de permettre à ces personnes de conserver leur domicile le plus longtemps possible.

1.1.3 ENVIRONNEMENTS INTELLIGENTS

Afin d'alléger le fardeau et l'effort requis pour subvenir aux besoins d'une personne en perte d'autonomie et lui permettre de maintenir son lieu de résidence le plus longtemps possible, plusieurs chercheurs tentent toujours d'incorporer des capteurs dans les lieux de résidences afin de supporter les personnes qui y habitent (Alam *et al.*, 2012). Ces habitats sont appelés environnements intelligents. Malgré qu'il soit irréaliste de penser qu'un environnement intelligent élimine complètement les besoins d'assistance d'une personne en perte d'autonomie, les environnements intelligents sont des outils qui ont le pouvoir d'augmenter le temps qu'une personne puisse conserver son lieu de résidence en plus d'alléger le fardeau des proches aidants.

Cook et Das (2004) définissent les environnements intelligents comme étant des environnements qui ont la capacité d'extraire et exploiter des informations de l'environnement dans le but d'améliorer l'expérience d'une personne dans cet environnement. Ici, deux concepts sont abordés, d'une part, l'accumulation de données issues des nombreux capteurs disposés dans l'environnement et de l'autre, l'intelligence ambiante.

En ce qui concerne l'accumulation d'information, un large spectre de capteurs peut être intégré dans un environnement pour la captation des mouvements d'une personne et des interactions de cette personne avec celui-ci. Plusieurs catégories de capteurs existent, notamment les capteurs bas niveau et haut niveau. Dans la catégorie bas niveau de ces capteurs, nous trouvons notamment les plaques de pression, les capteurs infrarouges passifs, les activateurs intelligents et les capteurs magnétiques. Cette famille de capteurs offre des informations binaires sur l'état de l'environnement (c'est-à-dire, s'ils sont activés ou non). Lorsque des interactions plus complexes veulent être mesurées, des capteurs plus haut niveau sont intégrés à l'environnement. Dans cette catégorie, nous trouvons, entre autres, les microphones et les caméras.

Les dernières années ont vu une diminution dans le coût et la taille des capteurs. Avec l'augmentation de l'autonomie énergétique des différents capteurs, un nouveau paradigme est apparu ; l'internet des objets (Lee et Lee, 2015). L'internet des objets désigne un réseau d'objets connectés à l'internet qui peuvent communiquer ensemble. Ce nouveau paradigme a démocratisé le développement d'habitats intelligents. Cette démocratisation a fait en sorte que les environnements intelligents sont de plus en plus courants : par exemple, l'utilisation

de téléphones intelligents et d'assistants intelligents tels que *Google Assistant*, *Siri* ou *Alexa*, pour contrôler et coordonner une suite de capteurs et d'actuateurs tels que des thermostats intelligents, détecteurs de présence, systèmes de sécurité, caméras, lumières connectées, interrupteurs intelligents, lecteurs de flux multimédias (tel que *Chromecast*), robots aspirateurs, etc.

En ce qui concerne l'aspect de l'intelligence ambiante, il s'agit en fait du coeur du problème du support. Afin d'assister une personne dans ses AVQ, il faut tout d'abord être capable de reconnaître ce que cette personne fait. Ceci réfère au problème de la reconnaissance d'activité.

1.1.4 RECONNAISSANCE D'ACTIVITÉS

La reconnaissance d'activités est un des plus grands défis pour le support d'une personne dans un environnement intelligent. La reconnaissance d'activités a pour but de déterminer l'action qu'un agent effectue dans un environnement à partir de données de capteurs (Sukthankar *et al.*, 2014). Les activités reconnues peuvent être très simples, comme reconnaître les activités *Sauter*, *Se pencher*, *Marcher* ou *Tomber*. Mais la complexité de la reconnaissance d'activités provient surtout des activités fondamentalement plus complexes, comme *Cuisiner*. *Cuisiner* est l'exemple parfait d'une activité complexe et qui est par le fait même une AVQ. Premièrement, *Cuisiner* est une activité complexe puisqu'elle est formée de plusieurs actions que l'on peut voir comme des sous-activités comme *Marcher*, *Prendre objet* et *Se pencher*. De plus, la variabilité entre deux instances est très grande. Cela s'explique par le fait qu'on ne

cuisine pas toujours le même plat et qu'on ne fait pas les mêmes gestes pour cuire un sachet de nouilles instantanées que pour préparer un rosbif. De surcroît, une grande variabilité existe également d'un environnement à l'autre. Cette variabilité est causée par la grande interaction avec l'environnement lors de l'activité. L'emplacement des différents objets, des meubles, des électroménagers et de l'espace de travail va nécessairement générer des mouvements différents d'un environnement à l'autre. De plus, l'activité se déroule généralement sur une longue période et elle peut facilement être entremêlée avec d'autres activités (comme *Laver Mains* et *Faire la Vaisselle*), entremêlement entre autres causé par les temps de cuisson. Ce sont ces grandes fluctuations qui rendent difficile l'extraction de caractéristiques fondamentales à l'activité. Puisque les AVQ sont généralement plus complexes, ils représentent donc un plus grand défi pour la reconnaissance d'activités.

La reconnaissance d'activités peut se diviser en deux familles d'approches ; l'approche exploitant des capteurs portatifs et l'approche exploitant des capteurs ambiants. En ce qui concerne l'approche par capteurs portatifs, celle-ci est axée sur l'utilisation de capteurs pouvant être portés par la personne dont on veut reconnaître les actions. Ces capteurs sont généralement des appareils disponibles sur le marché qui comportent déjà une suite de capteurs (principalement des accéléromètres) comme des montres intelligentes et des téléphones intelligents. Cette approche est utile pour faire la distinction entre certains mouvements répétitifs. Un exemple de cette approche est donné par Leonardis *et al.* (2018) où un accéléromètre triaxial, un gyroscope triaxial et un magnétomètre triaxial contenus dans un seul appareil et portés à la cuisse droite du participant sont utilisés pour faire la distinction entre 8 activités (soit

Assis, Debout, Couché, Marcher, Marcher (Pente Montante), Marcher (Pente Descendante), Monter Marches et Descendre Marches). Cette approche a comme avantage d'être facilement exploitable, peu importe l'environnement ; et la reconnaissance d'activités peut être réalisée, peu importe où se trouve la personne. Malheureusement, les informations que l'on peut extraire des différents capteurs dans cette approche sont souvent limitées et fluctuent en fonction de la position de l'appareil sur la personne (les mouvements enregistrés au poignet ne sont pas les mêmes qu'à la taille ou à la tête). De plus, les interactions avec l'environnement et les objets, partie importante des AVQ, ne sont pas mesurables avec cette approche. Finalement, il n'est également pas idéal de se fier à une personne dont les facultés cognitives sont en déclin pour porter un appareil supplémentaire.

L'autre approche, l'approche par capteurs ambiants, repose quant à elle sur des capteurs installés dans un environnement intelligent. Les capteurs utilisés dans cette approche sont variés. D'un côté, il est possible d'utiliser une suite de capteurs bas niveau (comme des interrupteurs intelligents, des plaques de pression, des capteurs magnétiques et des capteurs infrarouges passifs) intégrés dans l'environnement (Irvine *et al.*, 2019). L'état de ces capteurs à sortie binaire est observé dans le temps pour déterminer ce qui est fait dans l'environnement. Malgré que cette approche soit très peu intrusive, elle repose énormément sur l'intégration des capteurs dans l'environnement, ce qui peut laisser beaucoup d'angles morts. Par exemple, il faut plus qu'un capteur infrarouge passif placé dans la salle de bain pour déterminer ce qui s'y passe (il serait difficile de faire la distinction entre *Laver ses Mains*, *Faire ses besoins* et *Brosser ses Dents* sans capteurs supplémentaires). De l'autre côté, des capteurs beaucoup

plus haut niveau peuvent être installés dans l'environnement, comme des caméras et des microphones. Cette approche a comme avantage de pouvoir capter tout ce que la personne fait dans l'environnement, et même de percevoir les interactions avec l'environnement. Un exemple intéressant de cette approche est offert par les travaux de Madhuranga *et al.* (2020), où l'extraction de l'audio et du vidéo sont combinés pour faire la reconnaissance d'activités. L'utilisation de certains de ces capteurs peut cependant être très difficile à accepter, même dans le cadre de surveillance critique aux soins de santé (Offermann-van Heek *et al.*, 2019), puisqu'ils ont un impact négatif sur la vie privée.

Un type de radar a récemment généré beaucoup d'intérêt : les radars Ultra-Wideband (UWB). Les radars sont des appareils qui émettent des ondes radio, qui sont réfléchies par les surfaces puis sont captées à nouveau par le radar. Les réflexions permettent d'identifier la distance et même la vitesse des objets. Les radars UWB utilisent un large spectre radio (au moins 500 MHz) (Taylor, 2012). Cette particularité leur permet de percevoir des objets derrière les murs et autres surfaces, ce qui est idéal pour les environnements intelligents. Les radars UWB sont une alternative intéressante aux caméras, puisqu'ils offrent plus d'intimité, et les données sont assez explicites pour pouvoir mesurer les mouvements des objets et de la personne dans l'environnement. Malgré tout, cette approche a, elle aussi, ses limitations. Les radars UWB ont une portée limitée, et une précision qui diminue rapidement avec la distance, tout comme les caméras.

Nonobstant l'approche utilisée, les réseaux de neurones ont généré beaucoup d'intérêt dans les dernières années. Certaines architectures comme les réseaux de neurones convolutifs

se sont démarqués pour la résolution de problèmes de vision comme la reconnaissance d'objets (Zeiler et Fergus, 2013). Cela se reflète à travers les gagnants de défis de détection d'objets qui utilisent des réseaux de neurones convolutifs ou des variations de réseaux de neurones convolutifs, comme pour l'édition 2017 du défi de détection d'images d'*ImageNet* (*ImageNet Large Scale Visual Recognition Competition* (ILSVRC)) (Russakovsky *et al.*, 2015)¹. Cette approche fait contraste aux différentes approches plus conventionnelles comme l'utilisation de systèmes experts, qui sont basés sur l'exploitation de règles dérivées de connaissances d'experts dans le domaine d'application (Valenzuela *et al.*, 2004; Hatzilygeroudis et Prentzas, 2004).

1.2 PROBLÉMATIQUE DE RECHERCHE

Comme discuté plus haut, la reconnaissance d'activités est un problème de taille et qui est au centre de solutions pouvant assister des personnes dans la résolution de leurs AVQ quotidiennement. Dans la panoplie d'approches disponibles, l'utilisation de radars UWB semble la plus prometteuse puisque ces derniers offrent une bonne résolution et la capacité de percevoir derrière les objets (ce qui est particulièrement avantageux dans un environnement meublé contenant des divisions (murs)) et offre plus d'intimité que bien d'autres capteurs alternatifs (notamment les caméras).

1. Les résultats de l'édition 2017 du ILSVRC et une description sommaire des approches utilisées par chacune des équipes sont disponibles à l'adresse suivante : <http://image-net.org/challenges/LSVRC/2017/results>

Les travaux couverts par ce mémoire visent à atteindre un meilleur taux de reconnaissance d'activités de la vie quotidienne que les travaux précédents (notamment Maitre *et al.* (2021)) en combinant réseaux de neurones et radar UWB.

Plusieurs laboratoires ont comme centre d'activité l'élaboration de technologies ambiantes intelligentes. Au Québec, le laboratoire de *Domotique et informatique mobile à l'Université de Sherbrooke* (DOMUS) Giroux *et al.* (2008) et le *Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activités* (LIARA) situé à l'Université du Québec à Chicoutimi (UQAC) sont des exemples notables de laboratoires visant à développer des technologies de support reposant sur l'intelligence ambiante. Ce mémoire est réalisé au LIARA. Le LIARA a une longue expérience en ce qui concerne diverses solutions pour la reconnaissance d'activités (Maitre *et al.*, 2021; Fortin-Simard *et al.*, 2015; Maitre *et al.*, 2020), et autres technologies pour le support de personnes (Bouchard *et al.*, 2020a; Francillette *et al.*, 2018). Une contrainte implicite de ce mémoire est d'explorer des avenues qui n'ont pas encore été touchées par les travaux précédents du LIARA.

La réalisation de ce projet comporte bon nombre de défis. Dans un premier temps, une réflexion doit être faite sur les capteurs et les jeux de données utilisés. La quantité de données disponibles pour l'apprentissage des algorithmes est limitée à ce qui peut être enregistré au cours de ce projet de recherche et des expérimentations précédentes au LIARA. Un jeu de données limité peut affecter grandement la capacité des modèles à généraliser. L'année 2020 a été touchée par la COVID-19. Cet événement a fait en sorte de ralentir et même d'arrêter beaucoup de contacts interpersonnels et a ralenti certains services. Même si la COVID-19

fait obstruction à la capacité de bonifier de façon significative le jeu de données, cet aspect sera néanmoins abordé, car même s'il est difficile (voir impossible) de construire un jeu de données complet avec une foule de participants dans les délais et les contraintes imposées par la COVID-19, il est quand même possible d'utiliser des activités déjà enregistrées au LIARA dans le cadre d'expérimentations précédentes et de tester certaines pistes d'améliorations pour des expérimentations futures. Les détails et travaux concernant le jeu de données seront traités en plus amples détails dans la Section 3.2. Une caractérisation du comportement des radars UWB utilisés dans ce mémoire est aussi à envisager afin d'établir les limites de l'utilisation de ces capteurs.

Un autre défi provient du type d'activités traitées dans la littérature scientifique en ce qui concerne la reconnaissance d'activités avec radars UWB. La plupart des travaux se limitent à des activités très simples et à des environnements dépourvus d'obstacles (Jokanovic *et al.*, 2017; Du *et al.*, 2019). Une autre caractéristique de ce travail de recherche, en contraste avec la littérature scientifique, est l'utilisation de plusieurs capteurs au lieu d'un seul. Ces aspects font en sorte qu'une comparaison entre différents modèles est à faire pour déterminer quel type d'approche performe mieux dans ce contexte.

1.3 CONTRIBUTION DE LA RECHERCHE

Dans le cadre de la réalisation de ce travail de recherche, deux contributions principales sont apportées.

Dans un premier temps, deux approches sont explorées pour la reconnaissance d'activités. Une première approche est de tenter d'améliorer le taux de reconnaissance d'un réseau de neurones existant en intégrant une part d'expertise humaine sous la forme d'un système expert. Les limitations fondamentales de cette approche ont été très rapidement identifiées puis analysées. Puisqu'il s'agit d'une solution potentielle qui a été rapidement mise de côté, elle est couverte par ce mémoire, mais seulement de façon superficielle.

La deuxième approche consiste à simplement proposer un nouveau réseau de neurones visant à effectuer la reconnaissance d'activités à partir des données prétraitées de façon minimale. Cette approche combine deux classes d'architecture : les *Long Short-Term Memory* (LSTM) et les réseaux de neurones convolutifs (CNN). Cette approche exploite l'apprentissage par transfert, la fusion de données et une simple optimisation de la taille et la forme des filtres des couches CNN. Ce modèle, appelé *Tuned EfficientNetB0 avec LSTM*, offre une amélioration de 18,63 % du taux de reconnaissance par rapport aux réseaux de neurones développés dans des travaux antérieurs exploitant le même jeu de données.

Finalement, un jeu de données a été construit dans le but d'analyser les limitations du modèle sur la classification de certaines activités au LIARA.

1.4 MÉTHODOLOGIE DE LA RECHERCHE

Le mémoire présenté dans ce document a été réalisé en respectant une méthodologie scientifique divisée en 4 phases principales. La première phase consiste en l'acquisition de

connaissances dans le domaine de la reconnaissance d'activités, de l'apprentissage profond, des systèmes experts, des différents capteurs et des différentes approches utilisées pour la reconnaissance d'activités. Cette étape s'est faite sous la forme d'une revue de la littérature. Cette phase a été cruciale puisqu'elle a permis une meilleure compréhension du domaine en plus d'identifier les limitations des différentes solutions contenues dans l'état de l'art. Cela a permis également de guider les travaux réalisés.

La deuxième phase consiste en l'élaboration théorique d'un système de reconnaissance d'activités. Puisque ce travail s'inscrit dans l'ensemble des travaux réalisés au LIARA, l'approche proposée doit explorer à la fois de nouvelles avenues en ligne avec l'état de l'art en ce qui concerne la reconnaissance d'activités avec les radars UWB, et en plus être complémentaire aux approches des travaux antérieurs. De plus, l'objectif de tous les modèles développés est d'atteindre de meilleures performances que lesdits algorithmes développés précédemment.

La troisième phase consiste à mettre en application les modèles théoriques développés, incluant le prétraitement des données, la définition des divers modèles et l'entraînement de ces derniers. Le développement de la solution s'est fait avec le langage de programmation Python, combiné à l'utilisation des bibliothèques Keras et Tensorflow pour la modélisation et l'entraînement des modèles profonds.

La dernière phase consiste en la validation du modèle avec des données réelles. Pour ce faire, un jeu de données enregistré au LIARA a été utilisé pour mesurer la performance finale du modèle proposé. Les résultats sont disponibles à la Section 3.4.3, la Section 3.3.3 et

la Section 3.5.4.

1.5 ORGANISATION DU MÉMOIRE

Ce mémoire est composé de 4 chapitres. Le premier chapitre est une introduction à la problématique de recherche. Ce chapitre contient également une présentation sommaire des différents sujets touchés par le mémoire.

Le deuxième chapitre contient une revue de la littérature. Celle-ci couvre plusieurs sujets afin de clarifier les différents concepts touchés par les travaux présentés. Dans un premier temps, le fonctionnement des radars UWB sera expliqué. Ensuite, les différents concepts clés de l'apprentissage profond et son historique seront explorés. Une revue de quelques applications concrètes pour la reconnaissance d'activités avec radars UWB sera également réalisée. Ce chapitre permettra d'identifier les approches les plus prometteuses pour l'utilisation de l'apprentissage profond en combinaison avec des radars UWB.

Le troisième chapitre contient l'entièreté des travaux réalisés dans le cadre du mémoire. Dans un premier temps, une explication exhaustive du laboratoire LIARA et de la méthodologie utilisée pour mesurer les activités sera faite. Cela inclut un retour sur l'état du jeu de données tel qu'utilisé dans les travaux antérieurs, les expérimentations faites sur les capteurs, les améliorations apportées au jeu de données et la méthode utilisée pour traiter les données. Ensuite, le modèle final ainsi que les autres modèles considérés seront définis et leurs performances analysées. Un dernier aspect traité par ce chapitre est la création d'un nouveau jeu de

données et de l'analyse du modèle développé sur ce nouveau jeu de données.

Le dernier chapitre quant à lui sert à effectuer un retour général sur le mémoire en plus d'identifier les contributions et le potentiel de travaux futurs.

CHAPITRE II

REVUE DE LA LITTÉRATURE

Les travaux présentés dans ce mémoire se concentrent sur la question de la reconnaissance d'activités utilisant des capteurs de type radar UWB combinée à l'apprentissage profond. Il est ici important de faire un tour d'horizon des fondements de cette approche. Dans un premier temps, le fonctionnement des radars UWB sera expliqué suivi d'un survol des différentes techniques d'apprentissage profond touchées par ce mémoire. Certains cas précis d'utilisations des radars UWB, des applications de reconnaissance d'activités et d'utilisations de l'apprentissage profond seront, quant à eux, traitées dans la Section 2.2.

2.1 DÉFINITIONS

2.1.1 RADAR ULTRA-WIDEBAND

Un radar est un appareil électronique qui permet de mesurer la détection et la localisation d'objets en utilisant des ondes radio. Le mot radar vient de l'anglais et est dérivé de cette définition ; **R**adio **D**etection **A**nd **R**anging (Radar). Une onde radio a une fréquence variant entre 3Hz et 300GHz.

Lorsqu'une onde radio voyage dans un matériel tel que l'air et qu'elle entre en contact avec un autre matériel (ou une barrière) tel qu'une personne ou un avion, trois phénomènes peuvent survenir. Dans un premier cas, l'onde est réfléchiée, dans l'autre elle passe à travers et

dans le dernier, elle est absorbée. Dans le contexte d'un radar, le phénomène que l'on veut observer est le cas où l'onde radio est réfléchiée par l'objet. Plusieurs facteurs impactent la réflexion de l'onde. Dans un premier temps, la différence de matériaux est très importante. Ce phénomène est observable avec la différence de réflexion de la lumière (une autre onde électromagnétique) sur différentes surfaces comme un miroir et un tapis. Ensuite, la forme de l'objet impacte la façon dont les ondes sont réfléchies. Les avions furtifs utilisent généralement bien ces deux derniers points en utilisant des matériaux absorbants et des formes qui diminuent la quantité d'ondes radio réfléchies. Finalement, la longueur d'onde et le matériel de l'objet impactent comment l'onde est transmise à travers l'objet. De façon générale, plus la longueur d'onde est petite, plus elle sera atténuée lors de la transmission à travers le matériel.

Lorsqu'une onde entre en contact avec un objet, la vitesse de l'objet affecte également la fréquence de l'onde. Intuitivement, si un objet est statique, alors l'onde qui frappe l'objet et est retournée à la même fréquence. Si l'objet est en mouvement dans la direction de l'émission de l'onde, alors l'objet rentrera en contact plus fréquemment avec le front de l'onde, cette dernière sera donc retournée plus souvent, changeant ainsi la fréquence de l'onde lors de la réflexion pour une fréquence supérieure. À l'opposé, si l'objet est en direction opposée à la source de l'onde, alors l'objet rencontrera à une fréquence moins élevée les fronts d'onde, changeant ainsi également la fréquence de l'onde lors de la réflexion pour une fréquence inférieure. Ce phénomène de changement de fréquence est nommé effet Doppler (Ling, 2017). Un exemple commun est le changement de son d'une automobile. Lorsque celle-ci se dirige vers l'observateur, le son est plus aigu (compression de l'onde sonore) que lorsque la voiture

s'éloigne (dilatation de l'onde sonore). Ce phénomène est montré à la Figure 2.1. En observant la variation de la fréquence en retour, il est possible de déterminer la vitesse de l'objet.

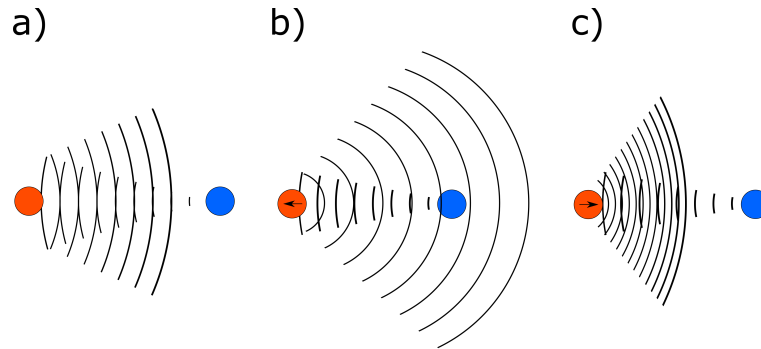


Figure 2.1 : Visualisation de l'effet Doppler : a) Effet Doppler sur un objet statique. b) Effet Doppler sur un objet s'éloignant de la source. c) Effet Doppler sur un objet se rapprochant de la source.

Un radar vise donc à exploiter la propriété réfléchissante des objets face aux ondes radio. Dans sa forme la plus simple, un radar est composé d'une antenne connectée à un oscillateur qui permet de générer un signal radio et d'une antenne connectée à un récepteur. Le récepteur mesure les variations apportées au signal d'origine pour déterminer la position d'objets. La portée d'un radar dépend de plusieurs facteurs incluant la puissance du transmetteur P_t , le gain de l'antenne du transmetteur G_t , la surface efficace de l'antenne du récepteur A_r (sa capacité à percevoir les ondes radio), la surface équivalente radar σ (simplement une mesure indiquant la détectabilité d'un objet par un radar, influencé par les matériaux et la forme de l'objet) et la distance R entre l'émetteur et l'objet observé. L'équation (2.1) qui donne la puissance au récepteur P_r montre la relation entre ces différents éléments (Li et Kiang, 2005) :

$$P_r = \frac{P_t G_t A_r \sigma}{(4\pi)^2 R^4} \quad (2.1)$$

La transmission d'onde radio peut se faire de deux façons. Il est possible de transmettre des ondes radio et de constamment en faire la réception. Il s'agit du mode continu d'opération. Ce mode d'opération permet seulement d'évaluer la vitesse d'un objet observé, puisqu'il est impossible de déterminer quand l'onde qui frappe l'objet a été émise et donc le temps qu'a pris l'onde à atteindre l'objet et de revenir. De plus, il est obligatoire pour ce genre de radars d'utiliser deux antennes, une pour la réception et l'autre pour l'émission puisque chaque antenne est constamment utilisée. Si le radar est opéré avec des *pulses*, c'est-à-dire qu'une onde électromagnétique est émise périodiquement, il est alors possible de mesurer le temps de réponse et donc de mesurer la distance de l'objet et en même temps la vitesse de celui-ci. Un avantage de ce type de radar est qu'une seule antenne est nécessaire, puisque l'émission et la réception ne se font pas simultanément.

Les radars Ultra-Wideband (UWB) (Taylor, 2000) (ou à bande ultra large en français) sont un type de radar qui a la particularité d'utiliser un large spectre de la bande radio allant jusqu'à 500 MHz, et opèrent avec une faible densité spectrale. Contrairement aux radars à bandes étroites qui utilisent seulement une fréquence de transmission et la conservent tout le long de la période de transmission, un radar UWB varie le signal transmis pour lui faire couvrir un large spectre de fréquences. Puisqu'il utilise une densité spectrale faible, ce type de radar a une portée limitée. Cette limitation n'empêche cependant pas son utilisation en

tant que radar, au contraire. L'utilisation d'un large spectre de fréquences permet au radar une grande précision pour des petits objets, voire même des parties d'un plus gros objet, et permet aussi de transmettre à travers des obstacles. Dans ce mémoire, les radars utilisés sont des radars UWB Doppler-Pulsés.

Les radars UWB ont un grand nombre d'applications. Dans un premier temps, leur précision permet de détecter certains signes vitaux sans contact sur une courte distance, comme la respiration et même la détection du pouls (Ren *et al.*, 2015). Un exemple de cette application est le radar Xethru X4M200 par Novelda² qui offre directement la mesure de la respiration. Il est intéressant ici de mentionner qu'il s'agit du modèle de radar exploité dans ce mémoire. Plus de détails sur l'utilisation de radars UWB pour la détection de signes vitaux sont donnés dans la Section 2.2.2. Un aspect important des radars UWB est leur capacité à pénétrer les objets, ce qui le rend idéal pour certains types d'applications comme l'assurance qualité (Cristofani *et al.*, 2016) où il est possible de détecter des irrégularités à l'intérieur d'un objet sans avoir à briser son intégrité. Cette capacité à percevoir derrière les objets est également exploitable lors d'opérations de recherche et de sauvetage (Jenssen *et al.*, 2018) dans le cas précis, par exemple, d'avalanches où il est possible de percevoir sous la neige. Même si les radars UWB ont connu un certain essor dans la littérature, le nombre de modèles de radars UWB ou de kits de développement construits autour de radars UWB est limité. Certains modèles ont été identifiés, comme les radars produits par Novelda (X4M200 et toutes

2. <https://www.sensorlogic.store/products/xe-thru-x4m200-respiration-sensor>

ses variantes³), ou encore certains kits de développement construits avec la même puce que le X4M200⁴, c'est à dire le système d'émetteur-récepteur radar à impulsion à courte portée sur puce X4 de Novelda. Ces modèles coûtent entre 250 USD et 3500 USD. Il est également possible de concevoir un radar UWB à partir de pièces déjà disponible (Malajner *et al.*, 2020), mais cette approche demande une certaine expertise.

La technologie UWB est également utilisée dans d'autres contextes, par exemple, le positionnement d'objets (Tiemann et Wietfeld, 2017). Dans ce scénario, les objets dont on veut mesurer la position sont munis de transmetteurs UWB qui émettent un signal de façon périodique et de multiples récepteurs (au minimum 3 pour permettre la triangulation) sont placés dans l'environnement. Ces récepteurs hautement synchronisés permettent de détecter des variations dans le temps d'arrivée de chaque transmission émise par le transmetteur et donc de trianguler la position du transmetteur. Cette approche est très utile puisque le signal peut plus facilement passer au travers des murs et de certains objets et permet une plus grande précision (10 cm) que d'autres approches utilisant d'autres protocoles sans-fil tel que Wifi (précision maximale entre 23 cm et 1 m) et Bluetooth (précision maximale entre 1 m et 3 m) (Zafari *et al.*, 2017).

Les données retournées par un radar UWB tel que le Xetheru X4M200 sont des données Distance-Doppler. Une donnée Distance-Doppler est une représentation des réflexions à différentes distances du radar. Un exemple de données Distance-Doppler retournées par un

3. <https://www.laonuri.com/en/product-category/novelda-en/x4-en/>

4. <https://www.sensorlogic.ai/chips-and-salsa/>

radar lors d'un *scan* est donné à la Figure 2.2. Dans cet exemple, il est possible de constater qu'il y a deux pics, le premier étant à 0,1 m de distance du radar et représentant le retour direct entre le transmetteur et le récepteur (peu importe la situation, ce pic sera présent dans les données) et le deuxième représentant un objet situé à approximativement 1,2 m devant le radar.

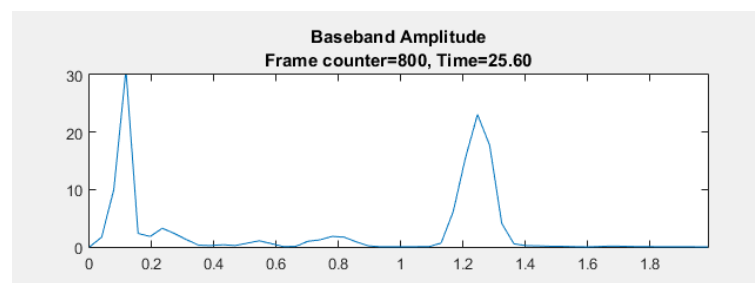


Figure 2.2 : Un retour de données représentant une donnée Distance-Doppler par un radar X4M200 tiré de Novelda AS (2018). L'axe horizontal représente la distance (en mètres) et l'axe vertical l'amplitude en dB.

Même si individuellement les données de radars UWB peuvent sembler simples, elles sont assez complexes lorsqu'il est question d'effectuer de la reconnaissance d'activités. Par exemple, la Figure 2.3 montre les données brutes extraites de trois radars UWB. Ces enregistrements ont été réalisés dans un habitat intelligent contenant 3 radars disposés à trois endroits différents dans l'environnement. Il est possible d'identifier quelques éléments. Premièrement, il est possible de constater que plusieurs objets se trouvent dans l'environnement. Ces objets sont représentés par les barres verticales. Il est également possible de constater que quelqu'un effectue des mouvements à quelques mètres du radar. Même s'il est possible de déterminer en un coup d'oeil que quelqu'un se trouve dans l'environnement, il est très difficile de déterminer

exactement quelle action est effectuée. Afin de déterminer l'action en cours, il faut extraire des *patterns* et des caractéristiques de ces variations de mouvements.

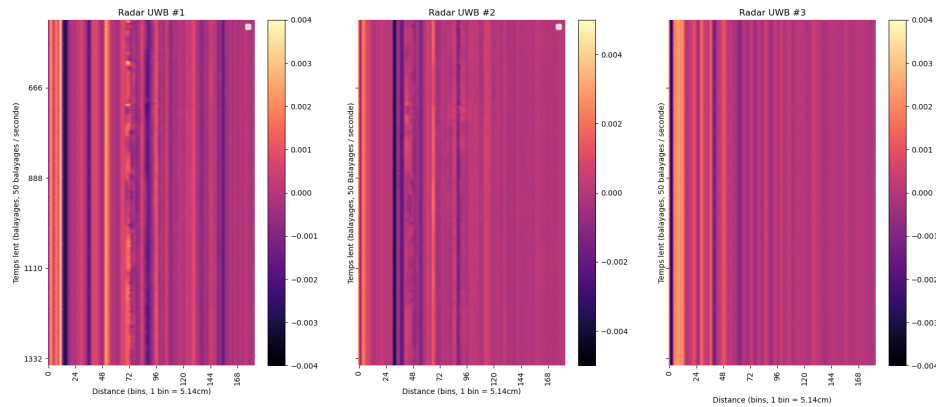


Figure 2.3 : Données brutes enregistrées lors de l'exécution d'une activité par une personne dans un habitat intelligent par 3 radars placés à des endroits différents.

2.1.2 APPRENTISSAGE PROFOND

Dans les dernières années, les réseaux de neurones ont fait des avancées spectaculaires en ce qui a trait aux problèmes complexes comme pour les problèmes de vision (Voulodimos *et al.*, 2018) telle la reconnaissance d'objets. Comme pour les problèmes de vision, les réseaux de neurones semblent être une bonne approche pour l'exploitation des données de radars UWB puisqu'ils peuvent extraire automatiquement les informations pertinentes d'un jeu de données (donc besoin minimal d'extraction de caractéristiques par un expert) et permettent même selon l'architecture choisie de trouver des *patterns* temporels, ce qui est particulièrement intéressant dans le cadre de la reconnaissance d'activités.

PERCEPTRON

Les réseaux de neurones artificiels sont des systèmes qui sont inspirés du fonctionnement du cerveau humain. Un cerveau est composé de neurones qui sont connectés entre eux par des liens synaptiques. Chaque neurone a la capacité de transmettre des signaux chimiques et électriques. De façon simplifiée et haut niveau, le raisonnement à l'intérieur du cerveau est simplement le passage de signaux à travers les neurones et les liens synaptiques. Un réseau de neurones est sensiblement similaire. Un réseau de neurones est composé d'une ou plusieurs unités (ou neurones), qui sont liées l'une à l'autre, et chaque unité est composée d'entrées, d'une fonction d'activation qui permet de calculer une sortie en fonction de la somme pondérée des entrées, et d'une sortie.

Quoique les réseaux de neurones aient eu un essor important dans les dernières années, les fondements théoriques datent quant à eux de 1943 (McCulloch, 1943). Ce n'est qu'en 1958 que Frank Rosenblatt propose le perceptron (Rosenblatt, 1958), l'unité de base du réseau de neurones artificiel. Au fil des avancées, nous arrivons donc au perceptron moderne tel qu'illustré à la Figure 2.4.

Chaque perceptron j possède une ou plusieurs entrées numériques $x_{j,i}$. Associé à chacune de ces entrées, un poids w_i est appliqué. Le poids permet d'ignorer ou d'accentuer une entrée en particulier. À l'intérieur du perceptron, les entrées sont multipliées par leur poids respectif puis additionnées. Le perceptron contient également une entrée de type biais b_j qui permet de

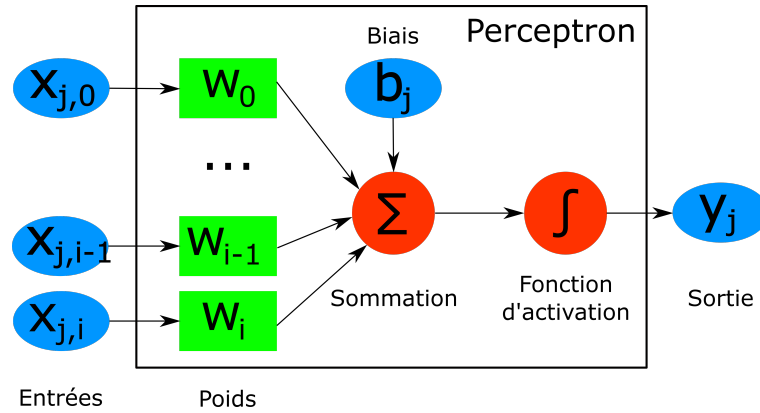


Figure 2.4 : Structure d'un perceptron

décaler le résultat de l'ensemble des entrées. Le biais est également additionné à l'ensemble des entrées pondérées. La valeur sommée est ensuite passée à une fonction d'activation. C'est la fonction d'activation qui détermine la sortie y_j du perceptron. Dans le perceptron tel qu'introduit par Rosenblatt, la fonction de Heaviside (ou fonction en marche d'escalier), comme définie à l'équation (2.2), est utilisée. Cette fonction d'activation permet une sortie binaire.

$$\forall x \in \mathbb{R}, H(x) = \begin{cases} 0 & \text{si } x < 0. \\ 1 & \text{si } x \geq 0. \end{cases} \quad (2.2)$$

Pour qu'un réseau de neurones opère comme désiré, il faut l'entraîner. L'entraînement d'un réseau de neurones est alors simplement l'entraînement de chaque perceptron. Lorsqu'on entraîne un perceptron, on vise à ajuster les poids sur chacune des entrées pour que lorsque l'on donne une certaine entrée au perceptron la sortie voulue soit donnée. La mise à jour des poids se fait en deux étapes, la première est le calcul de la sortie du perceptron décrit à

l'équation (2.3), la deuxième est l'ajustement des poids en fonction de la sortie du perceptron tel que décrit à l'équation (2.4). Lors du calcul de la sortie $y_j(t)$ d'un perceptron j avec un biais b à un instant t d'une entrée x_j appartenant à un jeu de données D , le résultat est donné par la fonction d'activation appliquée à la sommation des poids multipliés par leur entrée respective. Le poids mis à jour $w_i(t+1)$ quant à lui est le poids initial w_i additionné à la multiplication de la différence entre la sortie voulue \hat{y}_j et la sortie actuelle y_j , de l'entrée actuelle pour le poids $x_{j,i}$ et d'un taux d'apprentissage r . Le taux d'apprentissage permet d'éviter de trop gros changements de poids qui empêcheraient le perceptron de converger vers la réponse désirée.

$$y_j(t) = f\left(\sum_{i=0}^n (w_i(t) \cdot x_{j,i}) + b_j\right) \quad (2.3)$$

$$w_i(t+1) = w_i(t) + r \cdot (\hat{y}_j - y_j(t))x_{j,i} \quad (2.4)$$

L'apprentissage est réalisé jusqu'à ce que le comportement voulu soit observé à la sortie du perceptron ou que la différence entre la sortie voulue et la sortie actuelle soit assez petite pour les besoins du problème. Le terme *epoch* est utilisé pour dire que l'entraînement est fait sur l'entièreté du jeu de données. L'entraînement se fait généralement sur plusieurs *epochs*. Le perceptron tel que décrit plus haut est un classificateur linéaire.

Les perceptrons ne sont pas limités à utiliser les données comme entrées, ils peuvent aussi utiliser la sortie d'autres perceptrons. Pour cela, les perceptrons sont groupés dans des

couches qui sont elles-mêmes connectées entre elles. Une architecture de réseau de neurones est composée de ces couches de perceptrons qui se succèdent. On vise ici à reproduire les connexions des neurones dans le cerveau. Ces architectures basiques sont appelées des perceptrons multicouches ou réseaux de neurones à propagation avant. Il s'agit de la définition de facto d'un réseau de neurones. Un perceptron multicouche contient au minimum 3 couches (Hastie *et al.*, 2009). La première couche est une couche d'entrée. Cette couche sert simplement à relayer les données du problème au reste du modèle. La dernière couche est la couche de sortie et donne la sortie du modèle. Les couches se trouvant entre la couche d'entrée et la couche de sortie sont des couches cachées. C'est dans les couches cachées que la logique du modèle est située.

Ce genre de réseau de neurones effectue les calculs couche par couche, de la couche d'entrée vers la couche de sortie. Cette approche est appelée propagation avant. Il s'agit de la méthode la plus simple d'exécution d'un réseau de neurones.

Ces réseaux de neurones plus complexes utilisent une autre stratégie pour l'apprentissage. Dans le cas du perceptron, la descente du gradient est effectuée sur chacun des poids associés aux entrées. La descente du gradient est un algorithme d'optimisation. Cet algorithme est appliqué à une fonction de coût. La fonction de coût permet de quantifier comment le réseau de neurones répond au problème. Lors de l'entraînement, il faut réduire au minimum ce coût. Un coût idéal de 0 indique que la différence entre la sortie du réseau de neurones et les valeurs attendues est nulle. Plusieurs fonctions de coût peuvent être utilisées (notamment *Entropie Croisée*, *Erreur Absolue Moyenne*, *Erreur Quadratique Moyenne*, *Hinge*, *Huber* et *Kullback-*

Leibler). Une fonction de coût largement répandue est l'erreur quadratique moyenne (voir équation (2.5)). Dans le cas d'un réseau de neurones à propagation avant, la rétropropagation du gradient est effectuée (Werbos, 1974). De la même façon que l'on utilise la différence entre la sortie actuelle et voulue du perceptron pour ajuster les poids, le résultat d'une fonction de coût est partagé de couche en couche de la couche de sortie du réseau de neurones vers la couche d'entrée.

$$MSE = \frac{\sum_{i=0}^n (y_i - \hat{y}_i)^2}{n} \quad (2.5)$$

Il est important de noter ici que la convergence d'un réseau de neurones n'est pas garantie. Puisque les données sont utilisées pour faire l'apprentissage, si les données ne sont pas bien traitées en amont, ou si des différences entre deux entrées sont imperceptibles, alors le réseau de neurones ne devrait pas converger. De plus, certains problèmes requièrent des architectures spécifiques. C'est le cas de la reconnaissance d'images. Dans ce cas précis, la proximité des pixels contient beaucoup d'information sur le contenu de l'image, information qui est complètement perdue lorsque l'on aplatit l'image pour l'entrer dans la couche d'entrée d'un perceptron multicouche. De plus, il n'y a pas de règle spécifique concernant la taille et la profondeur requise pour répondre aux besoins d'un problème. Malheureusement, cette partie de la conception des architectures tient plus de l'instinct que de la science. Un réseau de neurones qui n'est pas assez profond ou trop étroit n'apprendra pas correctement (ou pas optimalement) alors que si le réseau de neurones est trop profond ou large alors il peut souffrir

de surapprentissage. Le surapprentissage survient lorsqu'un modèle finit par mémoriser les données d'apprentissage au lieu d'extraire le concept général derrière les données.

La fonction d'activation choisie pour chaque neurone impacte également fortement le comportement du réseau de neurones et de sa capacité à modéliser un problème. L'utilisation d'une fonction d'activation en marche d'escalier (*ON* et *OFF*) limite la capacité d'un ensemble de perceptrons à caractériser les subtilités derrière une entrée de données. Imaginons ici un réseau de neurones composé de quelques perceptrons avec une fonction d'activation en marche d'escalier qui a appris à extraire la couleur moyenne d'une image. Si la dernière couche est composée de trois perceptrons (représentant les canaux de couleurs rouge, vert et bleu), alors le réseau de neurones pourrait indiquer quelle couleur primaire ou secondaire représente la moyenne des couleurs (8 sorties possibles incluant blanc et noir), alors que si une fonction d'activation non binaire était utilisée (donc qu'une valeur entre 0 et 1 peut être donnée) alors il serait possible d'avoir une sortie qui estime une couleur en fonction de la concentration de chacun des canaux de couleur (assumant une sortie assignée sur 8-bit par canal, cela donne 2^{24} couleurs possibles). Des comportements plus complexes sont maintenant possibles grâce à des avancées dans les fonctions d'activations.

Deux des fonctions d'activation les plus répandues qui ont remplacé la fonction en marches d'escalier sont la fonction *sigmoïde* et la fonction *tangente hyperbolique*. Ces fonctions, définies respectivement aux équations (2.6) et (2.7), ont certains avantages. La fonction *sigmoïde* permet de facilement exprimer une probabilité entre 0 et 1 alors que la fonction *tangente hyperbolique* admet des valeurs négatives.

$$\text{sigm}(x) = (1 + e^{-x})^{-1} \quad (2.6)$$

$$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.7)$$

Un des problèmes qui peut survenir lors de l'apprentissage d'un réseau de neurones profond est la disparition du gradient lors de la rétropropagation du gradient (*Vanishing Gradient* (Aggarwal, 2018, p. 105-167)). Lorsque l'on propage le gradient, la valeur du gradient devient de plus en plus petite. Le problème survient lorsque cette valeur devient tellement petite que chaque poids ne varie plus, ce qui arrête l'apprentissage. Une des méthodes disponibles pour mitiger la disparition du gradient est l'utilisation de la fonction d'activation *Rectifier* (Glorot *et al.*, 2011) telle que définie à l'équation (2.8). Un neurone qui utilise la fonction *Rectifier* est appelé *Rectified Linear Unit* (ReLU).

$$\text{ReLU}(x) = \max(0, x) \quad (2.8)$$

Depuis leur introduction, *ReLU* est devenue la fonction d'activation la plus largement utilisée grâce au gain de performance que cette fonction a introduit dans les réseaux de neurones profonds (Ramachandran *et al.*, 2017a). Certaines variations de *ReLU* ont été introduites, mais peu ont eu autant d'attention que *swish* (Ramachandran *et al.*, 2017b). La fonction d'activation *swish* telle que définie à l'équation (2.9) a certains avantages par

rapport à *ReLU*. Contrairement à la fonction *ReLU* qui permet seulement des valeurs positives, la fonction *swish* permet d'avoir une petite valeur négative de sortie lorsque l'entrée est légèrement négative, mais une sortie près de 0 lorsque de grandes valeurs négatives sont données en entrée. Cela permet d'avoir un comportement similaire à *ReLU*, mais en plus les sorties négatives semblent permettre de mieux modéliser certains comportements. *Swish* a montré de meilleures performances sur plusieurs jeux de données, incluant la reconnaissance d'images par rapport à *ReLU* (Ramachandran *et al.*, 2017b).

$$swish(x) = x \cdot (1 + e^{-x})^{-1} \quad (2.9)$$

Avec le temps et l'augmentation des performances des ordinateurs, la taille des réseaux de neurones a augmenté. En augmentant le nombre de neurones par couche et le nombre total de couches, les réseaux de neurones sont capables de simuler des comportements de plus en plus complexes. Le terme *apprentissage profond* provient du nombre de couches cachées dans un réseau de neurones. Un réseau de neurones profond contient plusieurs couches cachées et est donc capable de raisonnement généralement plus complexe.

RÉSEAU DE NEURONES CONVOLUTIFS

Depuis leur introduction, plusieurs différentes classes de réseaux de neurones ainsi que plusieurs optimisations ont été introduites et font désormais partie du paysage de l'apprentissage profond. Ces approches cherchent généralement à répondre à une classe de problèmes en

particulier. De ces classes notables, nous trouvons les réseaux de neurones convolutifs. Cette classe de réseau de neurones a permis des avancées impressionnantes dans les problèmes de reconnaissance d'images. Les fondements ont été posés par Kunihiko Fukushima en 1980 avec ses travaux sur une architecture appelée *Neocognitron* (Fukushima, 1980). Ce réseau de neurones s'inspire du modèle du système nerveux visuel développé par Hubel et Wiesel (Hubel et Wiesel, 1959). Dans ce modèle, deux différents types de neurones se superposent en couches pour extraire les caractéristiques d'un stimulus visuel. Les neurones S permettent d'extraire des *patterns* alors que les neurones C permettent de combiner les informations de neurones S. La forme moderne des CNN découle quant à elle des travaux de Yann LeCun (LeCun *et al.*, 1990). LeCun est le premier à avoir intégré la rétropropagation du gradient pour entraîner les filtres, c'est-à-dire les matrices responsables de l'extraction des patterns visuels. Auparavant (comme pour *Neocognitron*), les filtres étaient sélectionnés manuellement. L'apprentissage des filtres facilite non seulement la création de CNN, mais les performances atteintes avec les filtres entraînés sont supérieures.

Un CNN est donc composé de la superposition de deux couches. Une de ces deux couches est la couche convolutive. Cette couche est composée de filtres. Les filtres ont une taille h par w et une profondeur d et sont composés de neurones. La profondeur d'un filtre est directement liée au nombre de canaux de l'image d'entrée, typiquement une profondeur de 1 pour les images en niveaux de gris et trois pour les images en couleur (*RGB*). Les filtres sont passés sur l'entièreté de l'image d'entrée avec un pas entre deux convolutions, et à chaque convolution, le produit scalaire de chacun des filtres est ajouté à la sortie. La Figure 2.5 montre

le principe de convolution appliqué à un et plusieurs filtres.

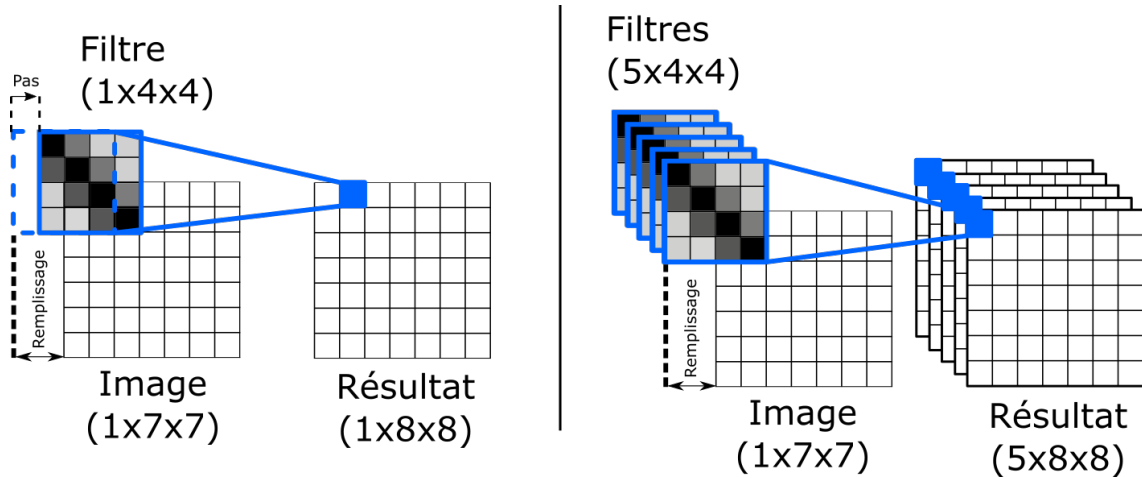


Figure 2.5 : Couche convolutive.

Lors de l'apprentissage, ce sont les poids de chacun des neurones des filtres qui sont ajustés. Ainsi, lors de l'apprentissage, le CNN apprend quelles configurations de filtres ou *patterns* de pixels permettent de minimiser l'erreur en sortie. Dans la couche convolutive, plusieurs filtres sont utilisés en parallèle, allant souvent jusqu'à 512 filtres par couche. Il est possible de calculer la taille de sortie de la couche convolutive W_{out} avec l'équation (2.10) où W_{in} représente la taille de l'entrée, K la taille du filtre, P le remplissage aux extrémités de l'entrée et S représente le pas entre deux convolutions. Ainsi, la sortie de la couche convolutive est beaucoup plus grande que l'entrée. Par exemple, une entrée d'image en niveaux de gris d'une hauteur de 5 et d'une largeur de 5 (couche d'entrée de 25 neurones) traitée par 8 filtres de taille 4 par 4, avec remplissage de 2 et un pas de 1 donne une sortie de taille de 8 de profondeur par 6 de hauteur par 6 de largeur (soit 288 neurones).

$$\frac{W_{in} - K + 2P}{S} + 1 = w_{out} \quad (2.10)$$

La deuxième couche importante des CNN est la couche de mise en commun. Comme discuté précédemment, la taille de sortie de la couche convolutive peut-être titanesque après quelques couches. Pour pallier à ce problème, des couches de mise en commun (*Pooling*) sont utilisées. Dans cette couche, une fenêtre de hauteur h et de largeur w est passée sur chacun des canaux d'entrée avec un pas de S et une fonction est appliquée sur cette fenêtre pour donner une valeur unique. La plupart du temps, il s'agit de la fonction max (*Max Pooling*). Le but de cette couche est de diminuer la taille du CNN à partir de cette couche. Le pas est donc souvent au moins aussi large que la taille de la fenêtre de mise en commun. Comme pour la couche de convolution, la taille de sortie de cette couche peut être calculée avec l'équation (2.10). Donc pour une couche d'entrée de $5 \times 4 \times 4$ (80 neurones) avec une fenêtre de 4×4 avec un pas de 4 et un remplissage de 2, la sortie sera donc de $5 \times 2 \times 2$ (20 neurones). La couche de mise en commun est décrite dans la Figure 2.6.

Les couches de convolution et de mise en commun permettent d'extraire des *patterns* visuels, mais sont rarement utilisées seules. Une architecture de CNN comporte également une couche d'aplatissement (couche transformant une couche multidimensionnelle en couche unidimensionnelle) suivi d'une (ou plusieurs) couche dense (couche de base utilisée dans un réseau de neurones à propagation avant où chaque neurone est connecté à tous les autres

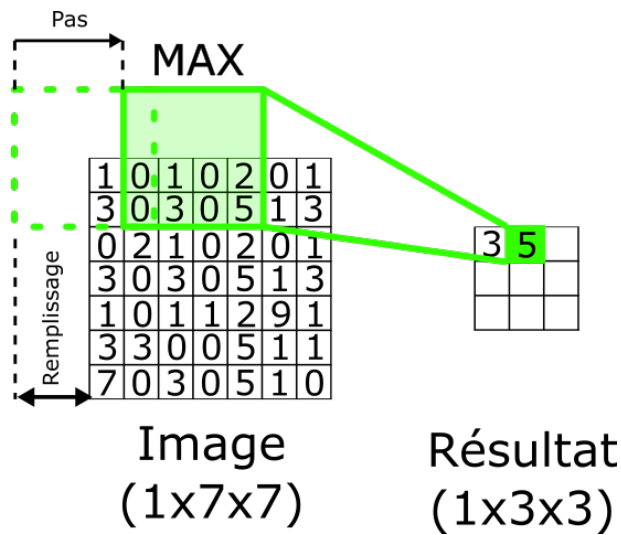


Figure 2.6 : Couche de mise en commun.

neurones de la couche suivante). Puisque les CNN sont utilisés dans le contexte de la reconnaissance d'image ou de détection, cette dernière couche sort souvent une classification ou une position de détection avec les dimensions d'une boîte de détection. La Figure 2.7 montre un exemple d'architecture complète d'un CNN.

Les CNN sont souvent très profonds. Un exemple notable est *Xception*, un des CNN les plus populaires. Ce dernier contient 36 couches de convolution et environ 23 millions de paramètres entraînaables (Chollet, 2016).

LONG SHORT-TERM MEMORY

Une autre classe de réseaux de neurones qui offre une différence intéressante par rapport aux réseaux de neurones classiques est le réseau de neurones récurrents (RNN) (Aggarwal,

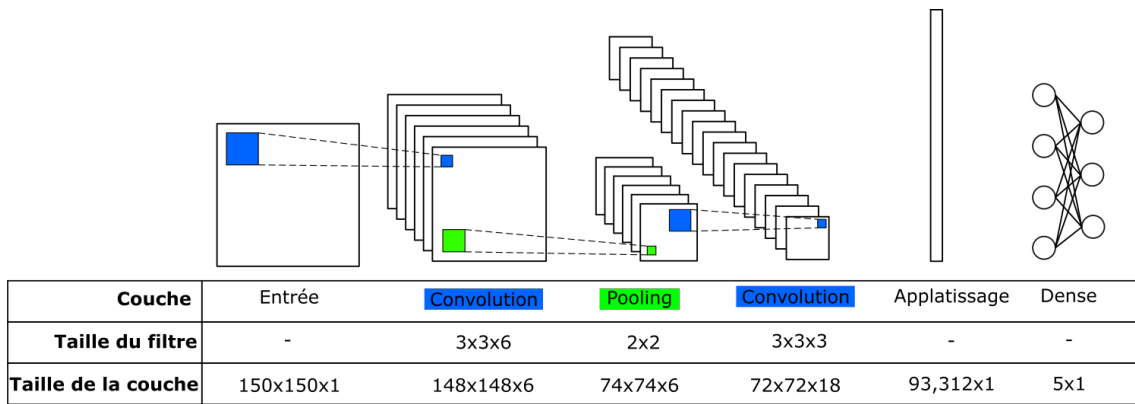


Figure 2.7 : Architecture d'un réseau de neurones convolutifs.

2018, p. 38-40). Dans un réseau de neurones classique, chaque entrée est considérée de façon séparée. Dans le cas des RNN, des cycles sont présents dans la structure. Ainsi, à un temps t , les neurones ont également accès à l'état du réseau de neurones au moment $t - 1$. Un exemple simplifié d'un cycle récursif est montré dans la Figure 2.8.

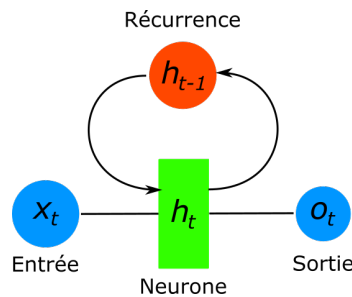


Figure 2.8 : Exemple simpliste d'un neurone avec cycle récursif.

C'est cette caractéristique qui permet à cette classe de réseaux de neurones de particulièrement bien performer dans des problèmes avec une forte corrélation au domaine temporel comme la prédiction de séries temporelles et la reconnaissance automatique de la parole (Graves *et al.*, 2013). Le *Long Short-Term Memory* (LSTM) (Hochreiter et Schmidhuber,

1997) est un des meilleurs (et plus répandus) exemples de RNN. Un neurone d'un LSTM contient une mémoire. La structure du neurone d'un LSTM est montrée à la Figure 2.9. Un neurone de LSTM a comme entrée la valeur actuelle de l'entrée en plus de la valeur précédente de sortie du neurone. Contrairement à un neurone de RNN cependant, le neurone du LSTM comporte un mécanisme pour conserver les valeurs pertinentes et d'ignorer les autres. Ce mécanisme est composé de trois portes : la porte d'oubli, d'entrée et de sortie. La Figure 2.9 montre les portes à l'intérieur d'un neurone LSTM, avec C comme valeur d'état, et h représentant les valeurs de sortie.

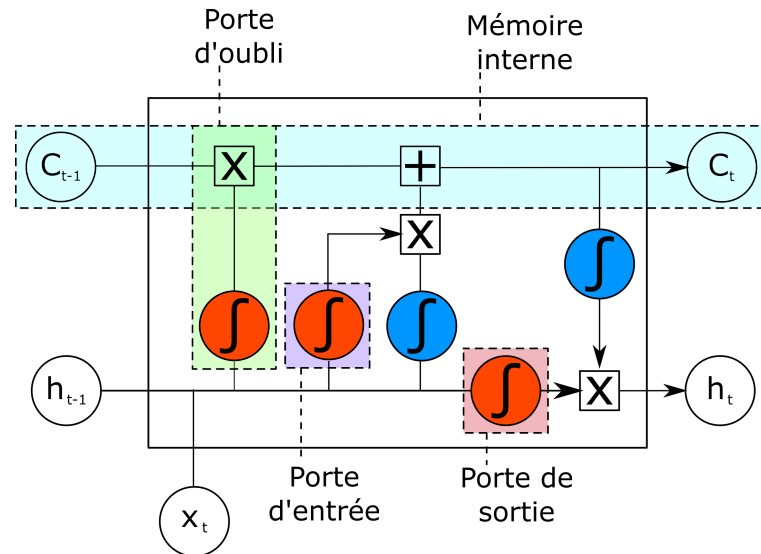


Figure 2.9 : Neurone de LSTM.

Dans un premier temps, la valeur d'entrée actuelle x_t est combinée à la sortie précédente du neurone h_{t-1} . Ces informations sont alors passées à la porte d'oubli. La porte d'oubli est simplement une fonction *sigmoïde* qui prend $[x_t, h_{t-1}]$ et qui effectue un produit matriciel de Hadamard sur la mémoire du neurone au temps précédent c_{t-1} . Cela permet d'oublier la valeur

de la mémoire interne lorsque la sortie de la fonction *sigmoïde* tend vers 0. Ensuite, l'entrée combinée à la sortie précédente $[x_t, h_{t-1}]$ est envoyée à la porte d'entrée. Cette porte permet de déterminer quelles informations doivent être conservées dans la mémoire du neurone. D'un côté, un état candidat est calculé avec une fonction *tangente hyperbolique*, et de l'autre, l'importance de l'information est calculée avec une fonction *sigmoïde*. Ces deux résultats sont combinés par un produit d'Hadamard et combinés à la mémoire avec une addition. Cette valeur de mémoire interne c_t sera donnée à la sortie du neurone. Finalement, l'entrée combinée à la sortie précédente $[x_t, h_{t-1}]$ est envoyée à l'entrée de la porte de sortie. La sortie du neurone h_t est calculée avec la mémoire actuelle c_t et l'entrée $[x_t, h_{t-1}]$. D'un côté, $[x_t, h_{t-1}]$ est donné à une fonction *sigmoïde*, permettant de déterminer quelles informations seront gardées à la sortie h_t , et de l'autre, la mémoire est passée par une fonction *tangente hyperbolique*. Ces deux valeurs sont multipliées avec un produit d'Hadamard et donnent h_t .

AUTO-ENCODEUR

Un dernier type de réseaux de neurones à considérer est les réseaux auto-encodeurs. Un réseau de neurones de type autoencodeur est une architecture d'apprentissage profond dont le but est d'encoder et de décoder les données d'entrées (Goodfellow *et al.*, 2016). Un autoencodeur est divisible en deux sous-modèles. Chacun de ces modèles est identique à l'autre, mais les couches sont dans le sens inverse. Ce genre de réseau est entraîné bout à bout dans le seul but de recréer les données en entrée. La première partie d'un autoencodeur est appelé "Encodeur" et transforme les données d'entrées sous une autre forme au centre

du modèle (souvent dans une forme compressée). La deuxième partie, appelée "Décodeur", retransforme la représentation des données d'entrées au centre du modèle à des données le plus proche possible des données d'entrées. La motivation de l'utilisation de ce genre d'architecture est que les caractéristiques importantes décrivant le mieux les données d'entrées se retrouvent au centre. Dans le cadre de l'utilisation d'autoencodeur pour la classification, seulement la partie encodeur est extraite du modèle et est combinée à quelques couches denses.

APPRENTISSAGE PAR TRANSFERT

Les réseaux de neurones qui atteignent les meilleures performances, notamment sur les problèmes de classification et de détection d'images, sont souvent très profonds (donc possèdent un nombre important de paramètres entraînaables) et sont entraînés sur des jeux de données de plus en plus gargantuesques. Il est intéressant de citer en exemple ici le jeu de données *OpenImages* (Kuznetsova *et al.*, 2020). *OpenImages* contient approximativement 9,2 millions d'images qui ont été étiquetées pour la classification, la détection et la segmentation, soit les trois grands problèmes de la reconnaissance d'images. Les réseaux de neurones profonds qui sont entraînés sur ce genre de jeu de données requièrent énormément de ressources et peuvent prendre jusqu'à quelques jours, voire une semaine de calcul ou plus selon les performances du matériel informatique utilisé. L'apprentissage par transfert (Ben-David et Schuller, 2003) vise à éliminer certains calculs redondants entre deux architectures identiques et deux problèmes semblables. Un réseau de neurones peut être divisé en deux parties ; l'épine dorsale, qui est composée de la majorité des couches inférieures et qui sont responsables de

la majorité du traitement des données d'entrée et la tête qui fait office soit de classeur ou de prédicteur. La distinction entre ces deux parties est d'autant plus visible dans les réseaux de neurones convolutifs où les couches convolutives responsables de l'extraction de *patterns* sont l'épine dorsale alors que les couches denses responsables de la classification représentent la tête. L'apprentissage par transfert consiste à transférer les poids des couches de l'épine dorsale déjà entraînées vers celles non entraînées et de geler ces couches pour qu'elles ne changent pas lors de l'entraînement sur le nouveau jeu de données. Ainsi, seulement la tête est entraînée et seulement une fraction des ressources de calculs est requise. Cela permet de diminuer énormément le temps de calcul.

LIMITATIONS

Malgré leur intérêt grandissant et le nombre d'applications qui augmente, les réseaux de neurones ont malgré tout quelques problèmes fondamentaux. Un des problèmes principaux est qu'un réseau de neurones est en somme une boîte noire, c'est-à-dire que le raisonnement effectué à l'intérieur est caché à l'utilisateur. En effet, même s'il est possible de visualiser chaque neurone et de mesurer comment chaque neurone répond en fonction d'une entrée, la logique globale du réseau de neurones reste incompréhensible. Même si plusieurs ont tenté d'extirper des connaissances symboliques des réseaux de neurones (Garcez *et al.*, 2001), les résultats restent encore limités et sont contraints à des architectures spécifiques. Il est donc impossible de connaître avec certitude le raisonnement derrière une décision. Dans le cas présent, pour un réseau de neurones entraîné pour la classification d'AVQ, il est donc

impossible de savoir pourquoi, à un moment donné, une AVQ a été priorisée par rapport à une autre.

Un autre grand problème des réseaux de neurones est, comme pour la plupart des algorithmes d'apprentissage machine, la sensibilité du réseau de neurones à la qualité des données. Il est non seulement primordial d'avoir suffisamment de données pour être capable d'extraire les caractéristiques générales d'un problème, mais il faut aussi être particulièrement attentif à l'introduction de biais dans les données. Quelques exemples notables existent et ont fait les manchettes, notamment l'application *Face Depixelizer* qui, lorsqu'il dépixelise le visage de l'ancien président américain Barack Obama, le rend blanc (Hamilton, 2020) ou encore un réseau de neurones dont le but est de prédire si une personne va commettre un crime seulement en analysant son visage (Harrisburg University of Science and Technology, 2020). Cette dernière application a été vivement critiquée, puisque ce type d'algorithme est entraîné sur des données qui affichent déjà des biais raciaux tirés directement des comportements sociaux (une personne provenant d'une minorité visible a plus de chance de se faire arrêter pour un crime comparativement à une personne blanche, ainsi qu'à purger des peines plus sévères (The Sentencing Project, 2018). Même si les biais ne sont pas toujours aussi apparents, ils peuvent exister dans n'importe quel jeu de données. Dans le cas de la reconnaissance d'AVQ, un jeu de données est constitué de différentes activités effectuées par différentes personnes. Dans ce cas en particulier, les activités n'ont pas toutes la même durée et ne sont probablement pas autant représentées. Par exemple, *Mettre des souliers* prend beaucoup moins de temps (donc moins de données sur lesquelles s'entraîner) et est effectué parfois moins

souvent que *Cuisiner*. Donc, si aucun effort n'est fait pour corriger le biais, un réseau de neurones entraîné sur ce jeu de données pourrait avoir tendance à prioriser l'AVQ *Cuisiner* au détriment de *Mettre des souliers*.

En plus de la qualité, la quantité de données disponible est également un facteur fondamental à la capacité d'un réseau de neurones à apprendre. Même si aucune règle mathématique n'existe pour déterminer la quantité de données exacte requise pour apprendre correctement, le nombre d'instances minimum par classe est généralement élevé. Par exemple, Alexey Bochkovskiy recommande environ 2000 instances d'images par classe que l'on veut détecter lors de l'entraînement de son de réseau de neurones pour détection d'objets *YOLO* (Bochkovskiy *et al.*, 2020).

Le même problème existe pour la variabilité des données au sein d'une même classe. Si les données ne sont pas assez variées, le modèle risque d'avoir des angles morts (il est impossible de couvrir des cas non présents dans les données d'apprentissage). Dans certains cas, il est facile d'avoir accès à une quantité importante de données, mais dans d'autres cas ce n'est pas aussi facile. Dans le cas d'un groupe de recherche sur la détection des AVQ qui exploitent les données de capteurs d'un appartement intelligent, la taille du jeu de données dépend du nombre de personnes qui acceptent d'effectuer des AVQ dans l'appartement et du temps qu'ils sont prêts à passer dans cet appartement intelligent. Il est donc peu probable que ce groupe puisse facilement enregistrer les activités de 2000 personnes dans l'appartement. Même s'il existe certaines techniques pour mitiger l'impact d'une quantité restreinte de données (techniques de *few shot learning* (Wang et Yao, 2019), qui veut littéralement dire

«apprentissage avec peu d’essais»), la construction du jeu de données de bonne taille et de bonne qualité est un des plus grands défis de l’utilisation de l’apprentissage profond.

2.1.3 SYSTÈMES EXPERTS

Les réseaux de neurones sont une approche basée sur les données, c’est-à-dire que la logique est extraite automatiquement des données sans intervention humaine. Plusieurs autres algorithmes utilisent cette approche, plus spécifiquement, dans les techniques d’apprentissage machine (Mitchell, 1997). Dans cette catégorie, nous trouvons notamment les machines à vecteurs de support (SVM) (Cortes et Vapnik, 1995) et les forêts d’arbres décisionnels (*Random Forest* (Ho, 1995)) qui sont souvent utilisées lors de la classification lorsqu’une foule de caractéristiques sont extraites des données. Un exemple est le travail de Bouchard *et al.* (2020b), où certains de ces algorithmes sont comparés pour la reconnaissance d’activités. Il existe un autre type d’approche, celle-ci basée sur les connaissances. Les porte-étendards de cette catégorie d’approche sont certainement les systèmes experts. Un système expert est un programme qui utilise des connaissances et un système d’inférence pour résoudre des problèmes assez complexes pour requérir de l’expertise humaine pour leurs solutions (Giarratano, 2005). La structure de base d’un système expert est composée de deux parties principales : une base de connaissances et un système d’inférence. La base de connaissances contient les informations sur le domaine d’application (appelées faits) et les règles qui le régissent. Le système d’inférence quant à lui sert à appliquer les règles définies dans la base de connaissances afin de trouver de nouveaux faits. La logique utilisée dans les systèmes experts

est généralement la logique du premier ordre (Cowell *et al.*, 1998). Le mode d'opération d'un système expert est simple, une requête (l'état d'un fait) est faite et le système d'inférence s'exécute jusqu'à la réponse soit trouvée (ou non). L'intérêt principal du système expert est qu'il est possible de facilement comprendre le raisonnement derrière une réponse à une demande. Les systèmes experts font partie des plus vieilles applications de l'intelligence artificielle et datent des années 1970 (Russell et Norvig, 2010). Un des aspects les plus ambivalents de systèmes experts est justement la contribution de l'expert. D'un côté, les connaissances d'un expert peuvent être difficiles à représenter dans un format autre que celui du système expert. De l'autre, la création de la base de connaissances et de ses règles peut être très laborieuse à créer et à maintenir, surtout si le domaine est sujet à changements.

2.2 CAS APPLIQUÉS

2.2.1 RECONNAISSANCE D'AVQ

Dans ce mémoire, l'utilisation des radars UWB est à l'étude, mais plusieurs autres approches existent pour effectuer la reconnaissance d'activités. Il est donc intéressant ici de faire un tour d'horizon des approches pour identifier où s'inscrit la reconnaissance à l'aide de radars UWB. En ce qui concerne la reconnaissance d'activités, plusieurs approches existent. Ces approches sont fondamentalement liées aux types et à la quantité de capteurs utilisés. Lorsque l'on compare la performance d'un classifieur, on utilise généralement le taux de reconnaissance. Le taux de reconnaissance est le nombre de classifications correctes divisé par le nombre total de classifications. Dans les cas où les classes ne sont pas équilibrées (les instances d'une ou

plusieurs classes sont en surnombre par rapport aux autres), le taux de reconnaissance balancé peut être utilisé. Ce dernier est calculé en calculant le taux de reconnaissance pour chacune des classes, puis en faisant la moyenne de ces taux de reconnaissances.

APPROCHES BASÉES SUR LES CAPTEURS PORTATIFS

Il est possible de diviser ces approches en deux catégories distinctes, soit les approches basées sur les capteurs portatifs et celles sur les capteurs ambiants. Dans la première catégorie, des capteurs qui peuvent être portés par la personne en observation sont utilisés pour reconnaître les activités. Il s'agit souvent d'appareils déjà disponibles sur le marché, comme des téléphones intelligents, des montres intelligentes et même des lunettes intelligentes. Ces appareils sont munis d'une panoplie de capteurs, principalement d'accéléromètres.

Une application possible de cette approche est caractérisée par Brunner *et al.* (2019). Dans ce travail, les chercheurs utilisent une montre intelligente pour différencier 5 styles de nage différents. Pour ce faire, 17 heures d'enregistrements ont été récoltées par 40 nageurs portant une montre intelligente disponible sur le marché. Les données de l'accéléromètre, du gyroscope, du magnétomètre, du baromètre et du capteur de lumière présents sur la montre sont utilisées. Les données sont simplement normalisées avant d'être utilisées par un CNN. Cette approche offre de bonnes performances avec un taux de reconnaissance de 96,4 %. Quoique ce type d'activité soit très répétitive et le nombre de classes limité, cela montre la capacité des CNN à extraire des *patterns* temporels à partir de données.

L'utilisation de capteurs portatifs est également applicable à des contextes où les activités sont plus complexes. Dans Nisar *et al.* (2020) par exemple, les mouvements enregistrés par un téléphone intelligent, une montre intelligente et des lunettes intelligentes sont analysés pour reconnaître 61 différentes activités atomiques (atomiques dans le sens d'indivisibles comme *Marcher, Se lever, Boire et Tourner*). Des caractéristiques sont extraites des capteurs, puis un SVM est utilisé pour la classification. En analysant la répartition temporelle de ces activités avec l'algorithme *Rank Pooling*, ils ont été capables d'atteindre une précision respectable de 63,64 % pour la classification de 7 différentes activités composites. Cette approche est intéressante puisque le problème de la reconnaissance d'activités est divisé en deux parties, la reconnaissance de mouvements et la reconnaissance d'activités en fonction de la reconnaissance de mouvements dans le temps.

Un des principaux attraits de l'utilisation de capteurs portatifs est que cette approche demande une quantité limitée de mise en place. De plus, le fait que les capteurs sont situés sur la personne permet de faire la reconnaissance d'activités dans divers environnements, contrairement à l'approche ambiante où les capteurs sont fixés dans l'environnement. Malheureusement, cette approche n'est pas parfaite. Un des principaux problèmes avec cette approche est que le capteur doit absolument être porté par la personne pour fonctionner. Cet aspect est d'autant plus important lorsque l'on développe des solutions pour des personnes en perte d'autonomie. Il se peut donc que la personne oublie simplement de porter le capteur. Cette approche a aussi une autre limitation fondamentale qui est le manque d'information contextuelle à l'activité. Par exemple, si l'accéléromètre dans une montre intelligente est utilisé

pour mesurer les activités effectuées dans un environnement, il est impossible de déterminer la position exacte de la personne dans l'environnement et quel objet est manipulé (si jamais un objet est manipulé) à moins que d'autres capteurs spécifiques permettent de mesurer cet état.

APPROCHES BASÉES SUR LES CAPTEURS AMBIANTS

Dans les approches basées sur les capteurs ambiants, des capteurs sont placés à des endroits stratégiques dans l'environnement afin de détecter les actions des personnes. Dans ce type d'approches, une foule de capteurs peuvent être utilisés. Dans la partie bas niveau du spectre des capteurs potentiels, on retrouve notamment les capteurs infrarouges passifs (PIR), les capteurs magnétiques et les interrupteurs intelligents. Dans ce cas en particulier, c'est l'interaction entre la personne et l'environnement qui est mesurée. L'arrivée de l'internet des objets (IoT) a eu un impact positif sur l'intérêt de ces approches, puisque la disponibilité des objets connectés a augmenté et leur prix diminué.

Un exemple de l'utilisation de capteurs de bas niveau pour la reconnaissance d'activités est Irvine *et al.* (2019). Dans ce travail, le jeu de données utilisé est le même que celui utilisé pour la compétition *UCAmI Cup* (Espinilla *et al.*, 2018). Ce jeu de données contient les données de 30 capteurs avec sorties binaires incluant des interrupteurs, des plaques de pression et des capteurs infrarouges passifs placés dans un appartement intelligent où 12 activités ont été exécutées par un seul participant. Les données sont collectées toutes les secondes pour une période de 30 secondes. Les activités dans ce jeu de données incluent *Brosser ses*

dents, Manger et Prendre des médicaments. De l'apprentissage ensembliste est utilisé en combinaison avec plusieurs modèles perceptron multicouche pour effectuer la classification et atteint un taux de reconnaissance de 80,39 % lors de la classification. L'apprentissage ensembliste vise à utiliser un ensemble de modèles afin d'améliorer la performance de la classification (Rokach, 2009). Dans le cas présent, 4 modèles sont combinés, un entraîné sur les activités du matin, un entraîné sur les activités en milieu de journée, un entraîné sur les activités de soir et un dernier entraîné sur l'ensemble des activités.

Dans Sukor *et al.* (2018), un système expert est créé afin de reconnaître des AVQ. Le jeu de données utilisé est composé de données en provenance de 14 capteurs binaires installés à divers endroits dans un environnement, tels que dans certains appareils électroménagers (pour déterminer s'ils sont utilisés), sur certaines portes et sur la toilette. Les activités à l'étude sont *Préparer à dîner, Préparer à déjeuner, Préparer un breuvage, Sortir, Prendre une douche, Dormir et Utiliser la toilette.* Le logiciel *Protégé*⁵ produit par l'université de Stanford est utilisé pour générer une suite de règles pour faire la classification des différentes activités. La base de connaissances est formée d'interactions avec les différents capteurs. Chaque évènement contient les informations date-heure, identification du capteur et son état. Cette approche offre de bons résultats, avec un taux de reconnaissance allant jusqu'à 90,9 % pour la reconnaissance de certaines activités. Ce travail est indicateur que l'utilisation de connaissances humaines peut permettre de différencier deux activités semblables.

5. <https://protege.stanford.edu/>

Puisque la reconnaissance d'une AVQ est fondamentalement plus complexe que d'autres types d'activités plus simples, des approches utilisant des capteurs plus haut niveau sont souvent priorisées. Par exemple, plusieurs travaux utilisent des caméras vidéo. Afin de mitiger les problèmes liés à la vie privée, plusieurs de ces approches utilisent des caméras RGB-D, c'est-à-dire des caméras incluant la profondeur en plus de l'image en couleur comme avec la Microsoft Kinect⁶, et extraient les silhouettes à partir du canal de profondeur et les squelettes des personnes au lieu d'utiliser les images brutes (Ercolano *et al.*, 2017; Madhuranga *et al.*, 2020). Comme pour la majorité des problèmes de vision, les CNN sont largement utilisés pour donner du sens aux images.

Dans Ercolano *et al.* (2017), c'est justement le squelette qui est utilisé pour faire la reconnaissance d'activités. Le squelette est une représentation de la posture d'une personne. Ce que ces chercheurs proposent est l'utilisation d'une accumulation de poses dans le temps pour classifier les activités. Cette représentation est fournie à deux modèles développés. Le premier modèle est composé de différents LSTM qui sont associés à des sous-ensembles des données, et le deuxième est un CNN-LSTM. Ces deux modèles sont testés sur le *Cornell Activity Dataset* (CAD-60) qui contient 60 vidéos enregistrées avec une Microsoft Kinect (donc des vidéos RGB-D) qui comportent 12 activités incluant *Cuisiner*, *Écrire sur un tableau blanc* et *Travailler sur un ordinateur*. Les résultats sont excellents, atteignant un taux de reconnaissance de 93,3 % pour le multi-LSTM et un taux de reconnaissance de 95,4 % pour le modèle CNN-LSTM.

6. <https://developer.microsoft.com/en-us/windows/kinect/>

L'approche proposée par Madhuranga *et al.* (2020) est un exemple intéressant de fusion de différentes sources d'informations (images RGB-D et microphone) pour effectuer la reconnaissance d'activités. Pour ce faire, un jeu de données a été créé comportant 24 activités enregistrées avec une caméra de type Microsoft Kinect et d'un micro avec l'aide de 17 participants. Parmi ces activités, on retrouve notamment les activités *Dormir*, *Tomber*, *Manger*, *Boire*, *Marcher*, *Lire* et *Laver les mains*. Dans un premier temps, les silhouettes des participants sont extraites à partir du canal profondeur de la Microsoft Kinect utilisée. Un CNN et un LSTM sont utilisés pour faire la reconnaissance de 20 activités différentes. Une limitation importante de l'extraction de silhouettes est exposée, où seulement 74,95 % des silhouettes sont extraites des images avec l'algorithme utilisé, surtout dans des scénarios où les participants ont une forte interaction avec des objets (comme le lit lors de l'activité *Dormir*). Dans l'autre partie de la solution, les caractéristiques fréquentielles des données du microphone sont extraites à l'aide de la transformée de Fourier rapide (*Fast Fourier Transform* - FFT) (Heideman *et al.*, 1985) et regroupées en intervalles pour former 127 groupes représentant l'entièreté du spectre de fréquences. Pour cette partie, un simple réseau de neurones à propagation avant de 4 couches est utilisé. Les deux modèles sont ensuite combinés et une sortie commune est ajoutée au modèle. La combinaison des deux offre une performance intéressante de 86,67 %. De meilleurs résultats encore (88,2 %) sont obtenus en combinant les trois modèles développés (CNN 3D, CNN-LSTM et Audio).

En ce qui concerne les radars UWB, ils se trouvent dans cette catégorie. Quoiqu'ils soient largement utilisés pour la reconnaissance d'activités simples comme la détection de

chutes, l'utilisation de radars UWB pour la reconnaissance d'activités complexes comme les AVQ n'est pratiquement pas présente dans la littérature. Les applications diverses des radars UWB sont traitées de façon plus exhaustive dans la Section 2.2.2.

2.2.2 APPLICATION DES RADARS ULTRA-WIDEBAND.

Depuis leur introduction, les radars UWB sont devenus un capteur de choix pour la détection de différents types de mouvements chez l'humain. Comme discuté précédemment, c'est la précision des mesures possibles en plus de la capacité à percevoir derrière les obstacles qui rend ces capteurs si intéressants. Cette section contient une revue des applications des radars UWB dans la littérature scientifique. De plus, afin de traiter des applications récentes seulement, cette section ne contient que des articles scientifiques ayant été publiés lors des 5 dernières années (c'est-à-dire de 2015 à 2020). Il est important ici de mentionner qu'il ne s'agit pas d'une revue exhaustive des applications des radars UWB, mais seulement quelques exemples clés qui permettent d'explorer diverses applications liées à la santé et à la reconnaissance d'activités.

Puisqu'il en sera question à plusieurs reprises durant cette section, des noms précis sont donnés à certains éléments des données des radars UWB. Les données des radars UWB sous la forme de données distance-Doppler. L'axe horizontal représente le temps et l'axe vertical la réflexion en fonction du temps de réponse d'un *pulse* qui peut être interprété comme la distance. Dans la littérature, l'axe horizontal est appelé temps lent (*slow-time* en anglais), l'axe vertical est appelé temps rapide (*fast-time* en anglais) et la mesure à une distance précise dans

l'axe horizontal est appelée *bin*.

DÉTECTION DE SIGNES VITAUX

Dans certains cas, il est impossible ou peu souhaitable de fixer des capteurs directement sur une personne, par exemple, le cas de nouveau-nés prématurés qui sont particulièrement fragiles. Plusieurs articles proposent des algorithmes et des approches pour faire la détection de signes vitaux. À courte distance, les radars UWB ont une assez bonne précision et un ratio signal sur bruit assez fort pour pouvoir identifier des variations infimes de mouvement du thorax. Cette caractéristique les rend idéaux pour la détection de signes vitaux, comme la respiration et le rythme cardiaque. La détection du rythme cardiaque est un véritable défi puisque les variations de mouvement du thorax sont très petites, de l'ordre de 0,2 mm. Ce genre d'application requiert donc soit un radar UWB avec une grande précision (et un très haut rythme d'acquisition de données) ou des traitements algorithmiques novateurs pour identifier les modulations cardiaques sous les modulations de la respiration.

Dans Ren *et al.* (2015), le but est de mesurer la fréquence de respiration d'une personne ainsi que son rythme cardiaque. Pour ce faire, deux méthodes sont comparées. Dans un premier temps, la méthode de démodulation de signal complexe est testée en appliquant la FFT au domaine complexe, et dans un deuxième temps en appliquant la méthode de démodulation arctangente (Steger *et al.*, 2006) au ratio des composants réels et imaginaires des *pulses* du radar UWB suivi d'une FFT. Ces méthodes ont été testées sur un sujet assis à 1,2 m puis 2

m devant le radar UWB opérant à une fréquence d'échantillonnage de 75 Hz. La méthode arctangente offre un plus grand ratio signal sur bruit que la méthode de démodulation de signal complexe, mais les deux ont d'excellents résultats avec 3,6 % d'erreur en comparaison avec le capteur fixé sur le participant à 1,2 m et 4 % d'erreur à 2 m.

Dans Ren *et al.* (2016), la variation de phase est mesurée afin de déterminer les mouvements effectués par le thorax d'une personne, et par le fait même, la respiration et le pouls. Cet article compare les méthodes de démodulation de signal complexe, démodulation arctangente, et leur variation respective avec représentation d'état. Pour ces méthodes, une FFT sur le temps lent est effectuée afin d'extraire la variation de phase. Les expérimentations sont effectuées avec un participant placé à 0,8 m du radar, ce qui est, relativement à d'autres travaux, une courte distance. Les résultats obtenus sont très bons avec une variation allant jusqu'à 1,5 % d'erreur type relativement à la mesure observée sur le participant pour les méthodes de démodulation de signal complexe avec représentation d'état et démodulation arctangente avec représentation d'état. Cette approche est intéressante puisque la variation de phase est généralement utilisée dans le contexte de radars UWB continus et est ici utilisée dans le contexte de radars UWB pulsés.

Dans l'approche proposée par Le *et al.* (2020), l'analyse spectrale singulière multivariée (MSSA) (Broomhead et King, 1986) est utilisée en combinant les signes vitaux à l'emplacement de la personne, c'est-à-dire au *bin* avec la plus grande amplitude, et sur les signes vitaux extraient avec la FFT sur le temps rapide. Cette approche offre de bons résultats avec 3,44 % d'erreur type avec l'appareil utilisé pour mesurer le rythme cardiaque de la personne.

DÉTECTION DE PERSONNES À TRAVERS LES MURS

La détection de la respiration peut être transposée dans un autre type de scénario. Puisqu'il est possible de détecter des mouvements derrière des murs et des obstacles, les radars UWB peuvent aussi être utilisés pour la détection de la respiration derrière un mur. Ce genre de technologie est particulièrement utile pour la recherche de survivants lors de catastrophes naturelles où des victimes peuvent se retrouver ensevelies sous des débris.

Une des implémentations les plus simples est apportée par Liang (2016). Le but de ce travail est de détecter la respiration d'une personne à travers différents obstacles. La respiration est un élément qui permet de différencier une personne d'un objet puisque le mouvement effectué lors de la respiration est perceptible sur des radars UWB. La respiration devrait donc apparaître dans le domaine fréquentiel puisqu'il s'agit d'un mouvement répétitif. Une comparaison est faite entre l'utilisation dans Singh *et al.* (2011) de la transformée discrète de Fourier (*Discrete Fourier Transform* - DFT) et de l'écart type des données traitées du radar. Pour les tests, un participant est placé à 1,83 m de distance du radar UWB et est séparé par un obstacle (un mur de 0,3 m de large en cloison sèche, ou une porte de bois dure). Avec la technique de l'écart type, la probabilité d'erreur de détection est nulle lorsque le seuil de détection est parfaitement ajusté.

L'approche proposée par Yan *et al.* (2016) se concentre sur la détection et la mesure de la respiration chez plusieurs participants à travers un mur. Dans un premier temps, les *bins* avec les plus grandes variances sont identifiés (indiquant la présence d'une personne). L'article

se concentre seulement sur les cas où les personnes sont à la même distance puisqu'il s'agit du cas le plus complexe. Ensuite, l'algorithme VMD (*Variational Mode Decomposition*) est utilisé pour extraire les modes du signal et donc les fréquences pertinentes. Finalement, la fréquence instantanée est calculée avec une transformée de Hilbert afin d'éliminer les fluctuations et le bruit. Cet algorithme est testé avec un radar UWB avec une fréquence de sortie de 152,6 Hz en temps lent et une précision de 4 mm sur 512 *bins*. Pour les expérimentations, la respiration de participants situés à 1,5 m de distance du radar et séparés par un mur est contrôlée. Cet algorithme a été testé avec 2 et 3 participants, et leurs respirations respectives ont été correctement mesurées par le radar UWB.

DÉTECTION DE CHUTES

Une des applications les plus évidentes des radars UWB dans un contexte de reconnaissance d'activités est la reconnaissance de chutes. Les chutes sont un évènement très important à surveiller chez une population d'un certain âge. Non seulement la fréquence de cet évènement augmente de façon considérable avec l'âge (Alshammari *et al.*, 2018), mais aussi puisque les conséquences peuvent être graves, pouvant causer fractures et divers problèmes limitant la capacité d'une personne à s'occuper d'elle-même (van Weel *et al.*, 1995). Ce problème de reconnaissance est une bonne porte d'entrée à la détection d'AVQ, puisque le problème contient un nombre limité de classes (seulement 2, chute et non-chute) et la forme de l'évènement (grand mouvement en peu de temps) facilite la tâche de détection par rapport à des activités plus complexes avec des mouvements plus petits qui se déroulent sur plusieurs

minutes.

Dans Ma *et al.* (2020), les chercheurs se concentrent sur la détection de chutes dans une pièce. Ils utilisent un réseau de neurones ConvLSTM. Il s'agit d'un réseau de neurones qui contient des couches convolutives et des unités LSTM. Cette architecture a l'avantage d'extraire des *patterns* locaux de chacun des pixels en plus des *patterns* temporels. Le modèle final contient deux couches convolutives 2D et une couche LSTM. L'approche a été testée dans un jeu de données composé de 6 activités (soit *Debout*, *Tomber*, *Couché*, *Se lever*, *Marcher* et *Sauter*) enregistrées par 5 participants. Le modèle est entraîné avec une validation croisée *Leave-One-Subject-Out* (LOSO), c'est-à-dire en mettant de côté les activités réalisées par une seule personne pour le jeu de test et de validation. Le modèle atteint un taux de reconnaissance moyen de 95,78 %. Un des tests notables présentés dans cet article est la performance de cette approche sur un environnement meublé. Cette expérimentation a montré une baisse notable de la performance, passant d'un taux de reconnaissance de 95,78 % à 78,3 %. Cela montre bien l'effet du changement d'environnement et des variations de lectures des radars UWB causées par la présence de meubles.

Une autre approche est de transformer les données distance-Doppler en série temporelle. C'est cette approche qui est utilisée par Sadreazami *et al.* (2020). Une fois les séries temporelles extraites, elles sont analysées par un CNN 1D. Ce modèle est testé sur un jeu de données composé de 2 activités (chute et non-chute) enregistrées dans un laboratoire à l'aide de 5 participants. Une validation croisée en un contre tous est faite et le modèle offre une performance de 92,72 %. Ce réseau de neurones performe mieux que les autres réseaux de

neurones testés, notamment un LSTM et un réseau de neurones à propagation avant.

Une autre approche qui transforme les données de radars UWB en série temporelle a été proposée par Maitre *et al.* (2020). Dans ce travail, la reconnaissance d'activités est effectuée dans un environnement intelligent contenant 3 radars UWB. Il est intéressant de mentionner ici qu'il s'agit de travaux qui ont également été réalisés au LIARA et donc utilisent les mêmes capteurs et le même environnement que celui utilisé dans les travaux contenus dans ce mémoire. Le jeu de données est constitué de quatre types de chutes et des non-chutes exécutées par 10 participants. Les quatre types de chutes sont *Chute avant*, *Chute de côté*, *Chute arrière* et *Chute en s'assoyant*. Les chutes sont effectuées à différents endroits dans l'environnement. Pour les non-chutes, il s'agit de segments d'AVQ (comme *Marcher*, *Faire le ménage* ou *Boire*) effectués sans contrainte. Cela aboutit en un jeu de données de 280 instances (segments de 15 secondes d'activité), 140 chutes et 140 non-chutes. Chaque instance est transformée en série temporelle. Pour ce faire, les valeurs dans un *scan* sont normalisées puis sommées afin de donner une seule valeur par *scan*. Une fois que tous les *scans* sont normalisés et sommés, l'ensemble des *scans* sont également normalisés. Puisqu'il y a 3 radars dans l'environnement, le même procédé est appliqué à tous les radars, et les trois sont combinés. Dans ce travail, un modèle CNN-LSTM est proposé. Ce modèle est composé de 5 CNN 1D. Chacun est nourri par un sous-ensemble de la série, et combiné à 2 couches de LSTM, puis une couche dense pour la classification. Cette approche offre une bonne performance, atteignant un taux de reconnaissance de 89 %.

RECONNAISSANCES D'ACTIVITÉS HUMAINES

Il est important ici de faire une distinction importante entre activités et AVQ. Dans la plupart des travaux sur le sujet de la reconnaissance d'activités, des activités minimalement complexes sont traitées (comme *Sauter*, *s'asseoir*, *Courir* et *Marcher*). Cette différence est majeure en comparaison avec les AVQ qui peuvent compter une suite importante de ces activités minimales (faire du thé peut requérir, par exemple, l'action se lever, marcher, s'accroupir, atteindre, etc.). Cette section est donc axée sur des mouvements plus simples à distinguer.

Jokanovic *et al.* (2017) par exemple, visent à reconnaître 4 différents mouvements. Le jeu de données est composé de 4 activités soit *S'asseoir*, *Tomber*, *Se pencher* et *Marcher* qui ont été réalisés par trois participants dans 3 orientations différentes devant un radar UWB. Leur approche consiste à diviser les données d'entrée en trois représentations différentes, soit le domaine distance-Doppler, fréquentiel et de distance. Ensuite, trois réseaux de neurones de type Auto-Encodeur sont entraînés et combinés à une couche dense pour classification. Ainsi, un modèle est créé pour chaque représentation des données. Un système de vote est ensuite ajouté pour déterminer quelle est la bonne classe en sortie. Les résultats sont très bons avec un taux de reconnaissance balancée de 96,4 %. Cette approche atteint de meilleurs résultats que chaque modèle pris individuellement.

Un autre exemple de reconnaissance d'activités combinant réseaux de neurones et radars UWB est proposé par Lang *et al.* (2017). Une particularité de ce travail est que le jeu de données

est composé de données simulées. Pour ce faire, des volumes représentant les différentes parties du corps sont associés aux différentes parties des squelettes du jeu de données MOCAP (*MOtion CAPture*) de l'université Carnegie Mellon. Ensuite, des données distance-Doppler sont générées avec un radar UWB simulé. Les activités reconnues sont *Marcher*, *Sauter*, *Courir*, *Boxer*, *Debout*, *Ramper* et *Marcher furtivement*. Le modèle développé est un CNN à trois couches. Le taux de reconnaissance atteint 98,34 %. Cette approche démontre la capacité des CNN à correctement différencier les différents mouvements. Cependant, le jeu de données utilisé ne comporte aucun bruit puisque les données sont simulées et donc parfaites.

Dans Yang *et al.* (2019), l'utilisation d'un réseau antagoniste génératif (*Generative Adversarial Network* (GAN)) est explorée. Un GAN est composé de deux réseaux, un générateur et un discriminateur, qui sont entraînés simultanément. Le générateur vise à générer des données qui se rapprochent le plus possible des données d'entrée, alors que le discriminateur vise à déterminer si une image a été générée ou fait partie du jeu de données. Le discriminateur est entraîné en alternant les images générées et faisant partie du jeu de données. Pour tester cette approche, deux jeux de données sont utilisés. Premièrement, de la même façon que dans Lang *et al.* (2017), des données distance-Doppler simulées à partir du jeu de données MOCAP. Ensuite, un deuxième jeu de données contenant des activités enregistrées par quatre participants, face à un radar UWB en laboratoire a été construit pour cette expérimentation. Les activités reconnues sont *Marcher*, *Sauter*, *Courir*, *Boxer*, *Debout*, *Ramper* et *Marcher furtivement*. Le modèle GAN utilisé est Open-GAN. Open-GAN contient des couches de convolutions et des couches denses. Cette approche offre d'excellents résultats, atteignant un

taux de reconnaissance de 97,03 % sur le jeu de données enregistré en laboratoire et 98,51 % sur le jeu de données MOCAP. Une des limitations principales de ce travail est que les activités enregistrées en laboratoire font toutes face au radar UWB. Très peu de variabilité existe donc entre les instances d'une même activité comme *Boxer*.

Dans un autre ordre d'idée, dans Ding *et al.* (2018), l'extraction de caractéristiques clés est combinée à l'utilisation d'algorithmes d'apprentissage machine. Le jeu de données utilisé est constitué de 12 activités divisées en deux catégories qui ont été effectuées par 13 participants. Dans la solution proposée, la première étape consiste à déterminer s'il s'agit d'une activité *in situ* (sur place) ou non. Cette distinction est faite à l'aide d'un SVM entraîné sur l'intervalle des *bins* touchés par l'activité. Pour chacun des deux types d'activités (*in situ* ou non), des caractéristiques différentes sont extraites et utilisées pour l'entraînement d'algorithmes d'apprentissage machine. Les meilleurs résultats atteints sont de 85,2 % avec l'algorithme K plus proches voisins (KNN) pour les activités *in situ* et 90,5 % avec l'algorithme *Bagged Trees* pour les activités non *in situ*.

De façon générale, il est intéressant de constater que la plupart des expérimentations se font dans une salle vide, près du radar UWB, ce qui permet d'obtenir des mesures beaucoup plus claires. Aucun des articles mentionnés précédemment n'a expérimenté dans des environnements complexes contenant des meubles et des murs comme dans un environnement réel.

2.2.3 TRAVAUX ANTÉRIEURS

Ce mémoire s’inscrit dans une lignée d’expérimentations pour la reconnaissance d’activités au LIARA. Considérant que ces expérimentations partagent un bon nombre de caractéristiques, incluant l’environnement, les capteurs et le jeu de données, il est intéressant d’analyser les approches utilisées afin d’identifier des pistes d’améliorations potentielles. De plus, comme le même jeu de données est utilisé, les modèles développés précédemment peuvent servir de point de comparaison pour mesurer l’impact des travaux.

Un premier travail intéressant réalisé au LIARA est celui de Bouchard *et al.* (2020b). L’approche utilisée est celle qui s’éloigne le plus de celle proposée. Elle est basée sur l’extraction de caractéristiques des données et l’utilisation d’algorithmes d’apprentissage machine pour faire la reconnaissance des activités. Dans un premier temps, les données sont transformées en séries temporelles. Ensuite, une foule de caractéristiques de ces séries temporelles sont extraites. Au total, 21 caractéristiques sont calculées par radar UWB pour un total de 63. Dans ce travail, 4 différents algorithmes sont comparés, KNN, forêt d’arbres décisionnels, arbre de décision et Adaboost. La meilleure performance est atteinte par la forêt d’arbres décisionnels avec un taux de reconnaissance de 80%. Il est important de mentionner ici qu’aucune validation croisée n’est faite (une séparation 70 / 30 est effectuée) et ne prend pas en compte la limitation du jeu de données. La limitation principale du jeu de données est le nombre restreint de personnes (10) ayant participé à sa création. Dans les cas où peu d’instances d’activités sont réalisées, il est avantageux conserver une instance d’activité entière pour le jeu de tests. Si ce n’est pas fait, des données qui sont très semblables se retrouvent dans le jeu d’entraînement

et dans le jeu de test.

Un autre travail antérieur intéressant est Maitre *et al.* (2021). Ce travail est également la suite logique de Maitre *et al.* (2020). Comme pour Maitre *et al.* (2020), une série temporelle est calculée en normalisant les valeurs dans un *scan* puis en les sommant afin de donner une seule valeur par *scan*. Une fois que tous les *scans* sont normalisés et sommés, l'ensemble des *scans* sont également normalisés. Puisque l'environnement contient 3 radars, cette opération est effectuée sur les données de chaque radar et les trois séries temporelles sont combinées. Pour ce travail, 4 modèles d'apprentissage profond sont comparés. Un modèle LSTM, un CNN-LSTM, ResNet et un modèle avec système de vote composé des trois autres modèles. Le modèle CNN-LSTM est constitué de différents CNN 1D combinés à des couches LSTM. Chaque réseau de neurones est nourri d'un sous-ensemble des données. En ce qui concerne le modèle LSTM, il s'agit simplement d'un réseau de neurones composé de couches LSTM et de couches denses. ResNet quant à lui, est un modèle CNN largement utilisé dans le contexte de la reconnaissance d'images (He *et al.*, 2015). Les résultats montrent que le meilleur modèle est le modèle avec système de vote avec un taux de reconnaissance 95 %, suivi de très près par le modèle CNN-LSTM avec un taux de reconnaissance de 94 %. Ces résultats sont effectués avec une séparation des instances 70 % d'entraînement, 15 % de validation et 15 % de test. Puisque les instances se superposent (un taux de superposition de 0,95 est utilisé lors de la génération du jeu de données), des données pratiquement identiques se retrouvent dans le jeu d'entraînement et de tests. Pour valider la généralisabilité du modèle CNN-LSTM, le modèle est entraîné avec validation croisée en *LOSO*. Les performances dans ce contexte sont

beaucoup moins élevées. Le modèle donne un Top-N. Dans cette représentation des résultats, il est montré si un modèle propose la bonne classe dans ses N premier choix. Donc un modèle qui offre un taux de reconnaissance de 50 % au Top-2, classifie correctement une instance dans ses deux choix les plus probables 50 % du temps. Dans ce scénario, le taux de reconnaissance du Top-1 (premier choix du modèle) varie en fonction du jeu de données (de la personne dans le jeu de test) variant entre 30 % et 43 %, mais atteint jusqu'à 90 % au Top-5. La généralisation, dans ce contexte, semble donc une caractéristique particulièrement importante à analyser.

CHAPITRE III

CRÉATION D'UN MODÈLE DE RECONNAISSANCE D'ACTIVITÉS

3.1 INTRODUCTION

Comme il a été décrit dans le chapitre 2 qui avait pour but d'identifier les approches proposées par la littérature, plusieurs approches sont disponibles pour la reconnaissance d'activités. Dans la plupart des cas, l'approche par apprentissage profond semble la plus prometteuse. Cette approche est principalement populaire par la simplicité de son utilisation dans des cas où des données complexes sont à traiter en plus de la qualité des résultats obtenus. Dans le cadre de ce mémoire, trois travaux principaux ont été réalisés. Dans un premier temps, la capacité des systèmes experts à améliorer les performances des modèles déjà entraînés a été analysée. Dans un deuxième temps, le développement d'un modèle d'apprentissage profond a été réalisé. Dans un dernier temps, la capacité des modèles à détecter différentes quantités de mouvement dans l'environnement intelligent du LIARA a été testée. Ce chapitre contient l'ensemble de ces travaux.

3.2 JEU DE DONNÉES

Dans le cadre de ce projet, les premières expérimentations ont été réalisées avec un jeu de données qui avait été construit au LIARA (Bouchard *et al.*, 2020b). Le LIARA possède un appartement intelligent qui contient un grand éventail de capteurs pour reconnaître des

activités. Pour l'enregistrement des activités, 3 radars UWB ont été placés dans l'appartement, couvrant au maximum l'espace de vie. Les radars sont placés à 36 cm du sol, soit la hauteur à laquelle les prises électriques sont généralement situées. Le plan de l'appartement intelligent du LIARA est montré à la Figure 3.1. Dans cette figure, les positions des radars UWB sont indiquées en rouge.

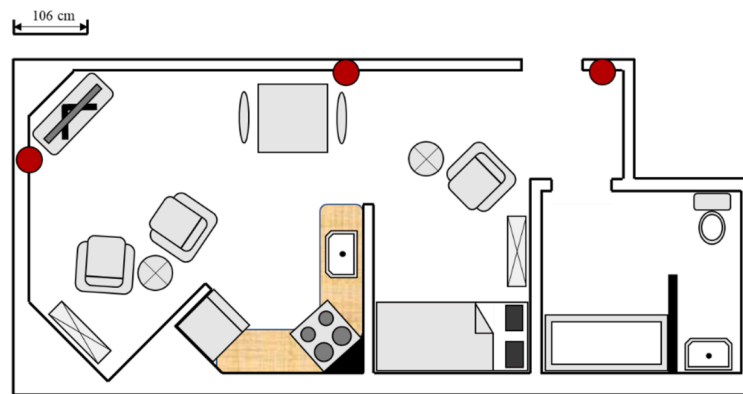


Figure 3.1 : Plan de l'appartement intelligent du LIARA. La position des radars UWB est indiquée en rouge. Cette figure est tirée de Maitre et al. (2021). Reproduit avec la permission de Elsevier.

Le jeu de données contient 15 activités distinctes ; *Laver ses dents*, *Faire le ménage*, *Cuisiner*, *Faire la vaisselle*, *Boire*, *Manger*, *Faire du thé*, *Enfiler une veste*, *Mettre des chaussures*, *Lire*, *Dormir*, *Prendre des médicaments*, *Utiliser un ordinateur*, *Marcher* et *Laver les mains*. 10 participants ont participé à l'enregistrement de ces activités. Aucune indication n'était fournie aux participants quant à la manière et le lieu où devait se réaliser l'activité. Il était donc à la discrétion des participants d'effectuer l'activité de la façon et à l'endroit qu'ils désiraient. Cela permet que la réalisation des activités soit réaliste et variée. Les activités enregistrées ont une durée variant de 15 secondes à 300 secondes. Les informations sur la longueur des

différentes classes d'activités sont comprises dans le Tableau 3.1.

Activité	Durée	Durée	Durée
	Minimum (s)	Maximum (s)	Moyenne (s)
Boire	15	28	26,6
Dormir	38	58	55,2
Enfiler une Veste	21	29	26,5
Faire le Ménage	92	118,00	115,0
Cuisiner	231	299	291,3
Faire du Thé	129	178	169
Faire la Vaisselle	98	118	114,9
Laver les Dents	120	179	171,8
Laver les Mains	21	29	26,8
Lire	97	118	114,2
Manger	76	118	113,4
Marcher	22	29	26,9
Mettre des Chaussures	27	43	40,7
Prendre des Médication	13	28	25,7
Utiliser un Ordinateur	93	118	114,5

Tableau 3.1 : Durées des différentes activités.

Une grande différence de durée de temps existe entre la plupart de ces activités. Cette variation est un point important, puisqu'il est possible d'extraire plus d'échantillons d'une plus longue instance d'activité que d'une courte. Il s'agit d'un point à surveiller, car lors de la création d'un jeu de données, il est possible que certaines activités soient plus représentées par rapport à d'autres, impactant la capacité d'un modèle d'apprentissage profond (et modèles

classiques) à correctement classer chacune des activités si aucune mesure n'est mise en place pour mitiger le problème.

Les radars UWB utilisés pour la captation des activités sont des Xethru X4M200 de Novelda⁷. Un des arguments de vente principaux de ce radar UWB est sa capacité à faire évaluer la respiration et cela sans contact à courte distance. Un autre argument de vente de ce radar est le prix relativement bas (250 \$ USD) qui le rend donc intéressant dans une perspective de création d'environnement intelligent. Ce radar opère entre 6,0 GHz et 10,2 GHz et est configuré pour opérer à des distances variant de 0,4 m à 9,85 m. Les informations fournies par le radar UWB sont des données Distance-Doppler, c'est-à-dire la quantité de réflexions mesurée en fonction de la distance. Les informations de distance retournées par le radar sont contenues dans un tableau contenant 184 *bins*. Chaque *bin* contient donc l'information d'un intervalle de 5,14 cm. Ce radar est configuré pour opérer à une fréquence de 50 Hz (soit *scans* par seconde). La Figure 2.2 montre un *scan* effectué en laboratoire qui est tiré directement de la fiche technique du radar (Novelda AS, 2018).

Un exemple d'un *scan* effectué par le radar UWB, mais cette fois-ci effectué dans le LIARA, est montré à la Figure 3.2.

Dans la Figure 2.2, un seul objet est placé devant le radar. Dans la Figure 3.2, il s'agit d'un environnement complexe complètement meublé. La différence entre les deux est significative et permet d'illustrer l'impact d'un environnement meublé sur l'acquisition de données lors de

7. <https://www.sensorlogic.store/products/xe-thru-x4m200-respiration-sensor>

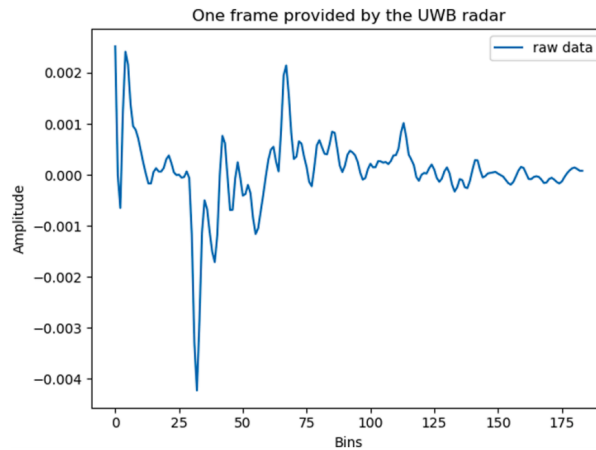


Figure 3.2 : Un *scan* capté par le Xetheru X4M200 (Distance-Doppler) au LIARA. Figure tirée de Maitre et al. (2021). Reproduit avec la permission de Elsevier.

la réalisation d'activités. Dans le premier cas, un objet seul est facilement détectable, alors que dans l'autre il est difficile de déterminer ce qui est présent dans l'environnement.

En plus des données Distance-Doppler fournies par les radars, la position de la personne dans l'environnement est estimée à l'aide d'un algorithme de trilatération (Alarifi *et al.*, 2016). L'algorithme exact utilisé est sujet au secret commercial, mais le fonctionnement général de la trilatération est tout de même respecté ; il est possible de déterminer la position d'un objet en comparant une mesure de distance entre l'objet observé et trois capteurs. Cela se fait en comparant la distance calculée entre l'objet et les points de référence. Une méthode largement utilisée pour estimer la distance de l'objet et d'un capteur est la conversion du délai de propagation d'ondes entre l'émetteur et l'objet en distance. La position est particulièrement difficile à estimer dans l'environnement intelligent du LIARA, car plusieurs objets sont dans le champ de vision de chaque radar UWB (voir Figure 3.1). Il y a donc une grande variation sur la position estimée de la personne. Il n'est donc pas judicieux de se baser purement sur

cette information pour classifier les différentes activités. La Figure 3.3 montre le principe de la trilatération.

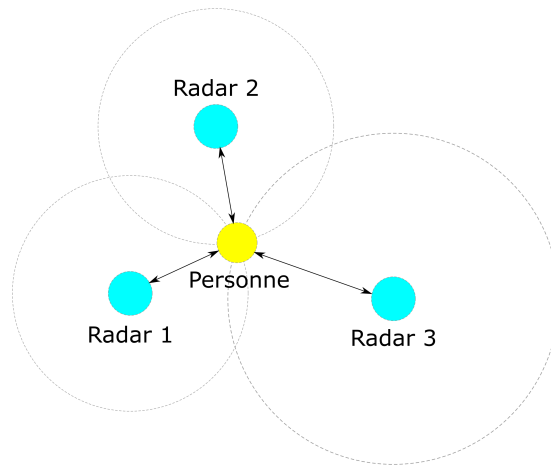


Figure 3.3 : Exemple visuel de la trilatération.

Contrairement à plusieurs jeux de données utilisés dans d'autres travaux, les données utilisées pour ce mémoire ont été récoltées dans un environnement semblable à un environnement de déploiement réel. Cet aspect particulièrement important. Puisque les données ont été récoltées dans un environnement meublé, il est plus que probable que des obstacles se trouvent entre les capteurs et la personne observée. Cela affecte donc la qualité des observations : du bruit (meubles et murs) sera inclus également dans les données. De plus, un aspect distinctif de ce jeu de données est la quantité d'activités complexes contenues dans ce dernier.

Un point faible cependant de ce jeu de données est le nombre restreint de participants. Comme discuté plus tôt dans la Section 2.1.2, le manque de données est un frein important à la généralisation des modèles. De plus, la différence entre deux instances d'une même

activité peut être grande dans ce jeu de données. Ainsi, même si un modèle est capable de très bien reconnaître l'activité présente dans le jeu de données, le manque de données pourrait affecter la capacité du modèle à reconnaître cette même activité, mais effectuée par une nouvelle personne. Cette capacité à utiliser des données non présentes dans le jeu d'entraînement, appelée généralisation, permet de démontrer si le modèle est capable d'extraire les caractéristiques fondamentales des classes à identifier et non d'extraire les particularités des instances de ces classes. Avec un jeu de données restreint, il est beaucoup plus difficile de développer un modèle avec une bonne capacité à généraliser.

3.3 EXPLORATION DE L'UTILISATION D'UN SYSTÈMES EXPERTS

Les premières expérimentations effectuées dans le cadre de ce projet de recherche concernent la combinaison de modèles déjà existants avec un système expert. Un modèle précédemment développé au LIARA (Maitre *et al.*, 2021) n'offre pas une prédiction particulièrement élevée lorsque la validation croisée de type *Leave-One-Subject-Out* (LOSO) est utilisée. Dans la validation croisée LOSO, les activités réalisées par un des participants sont mises de côté pour le jeu de test, alors que le reste est utilisé pour l'entraînement. Ainsi, il est possible de déterminer si le modèle réagit bien face à des données sur lesquelles il ne s'est pas entraîné. Cette technique est particulièrement utile pour tester un aspect précis de la généralisation, dans ce cas-ci les personnes. Dans le cas présent, 10 jeux de données LOSO sont construits puisque 10 participants ont participé aux enregistrements. De plus, les résultats sont donnés avec un taux de reconnaissance au Top-N. Avec les modèles d'apprentissage

profond, une probabilité d'appartenance à chacune des classes est donnée au lieu d'une réponse unique, ce qui permet facilement l'utilisation de cette mesure. Pour en revenir au modèle proposé dans les travaux précédents du LIARA, son taux de reconnaissance moyen pour le Top-1 se situe approximativement à 46,96 %. Cependant, le taux de reconnaissance augmente dramatiquement jusqu'au Top-5, qui atteint en moyenne 86,33 %.

Le modèle proposé précédemment se base seulement sur une série temporelle dérivée des données brutes de radars UWB. Aucune connaissance experte n'est donc utilisée pour faciliter la classification. La question qui est posée est donc la suivante : *est-il possible d'augmenter la capacité de ce modèle à classifier correctement les activités en le combinant à un système expert ?* L'idée ici est donc d'utiliser les résultats de la classification de ce modèle et de le combiner à des règles logiques développées à partir de caractéristiques extraites des données afin d'aider à la classification.

3.3.1 MODÈLE DÉVELOPPÉ DANS LES TRAVAUX ANTÉRIEURS

Le modèle utilisé comme classeur principal est un réseau de neurones qui a été développé dans le cadre de travaux précédents (Maitre *et al.*, 2021). Il s'agit d'un modèle combinant de multiples CNN 1D à des couches LSTM. Dans un premier temps, les données des capteurs sont transformées en série temporelle. Le modèle exploite des instances de données enregistrées par les radars UWB de 15 secondes. Plus précisément, l'ensemble des acquisitions ont été segmentées à l'aide d'une fenêtre glissante de 15 secondes avec un chevauchement de 80 % entre deux fenêtres successives. La transformation d'un tableau de données Distance-Doppler

en série temporelle se fait de la façon suivante ; les valeurs dans un *scan* sont normalisées puis sommées afin de donner une seule valeur par *scan*. Une fois que tous les *scans* sont normalisés et sommés, l'ensemble des *scans* sont également normalisés. Puisqu'il y a 3 radars dans l'environnement, le même procédé est appliqué à tous les radars, et les trois sont combinés.

Le modèle est composé de 5 CNN qui sont assignés à un sous-ensemble de la série temporelle. Ces CNN sont par la suite combinés à de multiples couches de LSTM. Des couches denses sont ajoutées à la fin du modèle pour effectuer la classification. Afin de tester la généralisation du modèle, le modèle est entraîné sur le jeu de données avec validation croisée en LOSO. L'architecture du modèle tel que d'écrit dans Maitre *et al.* (2021) est illustré à la Figure 3.4. Pour plus de détail sur l'architecture, référez à Maitre *et al.* (2021).

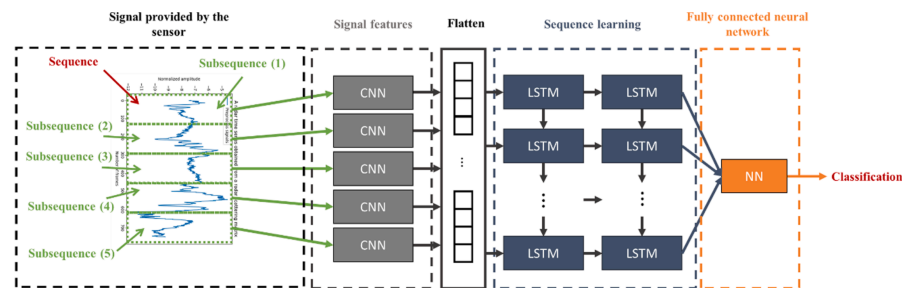


Figure 3.4 : Architecture du modèle CNN-LSTM. Figure tirée de Maitre et al. (2021). Reproduit avec la permission de Elsevier.

3.3.2 SYSTÈME EXPERT

Le but de cette expérimentation était de vérifier s'il était possible d'améliorer les performances d'un modèle existant en ajoutant une couche de connaissance experte sur le

domaine. Il s'agit d'une expérimentation qui est pertinente dans le contexte de la recherche au LIARA. Puisque le système expert est construit dans le but d'optimiser ce modèle, celui-ci est appelé «modèle central» dans le reste du mémoire.

La combinaison du système expert avec les réseaux de neurones n'a pas donné lieu à des résultats prometteurs. Nous avons décidé de tout de même décrire brièvement les expérimentations afin de documenter cette approche d'intelligence artificielle classique pour les prochains chercheurs qui auraient une idée similaire. Elle est donc volontairement très succincte, mais repose sur les fondements classiques de l'intelligence artificielle (Russell et Norvig, 2010).

La solution finale est inspirée de deux concepts. D'une part, le fonctionnement général de l'application est basé sur la structure des agents intelligents. Les éléments principaux de l'architecture des agents intelligents sont les suivants : perception du monde, mise à jour d'une représentation interne du monde, calcul d'une action à apporter et mise en performance de l'action dans le monde. Dans le cas présent, la perception du monde est faite par les radars UWB, la représentation du monde est contenue dans le système expert et l'activité reconnue représente l'action sur le monde. Cette architecture a plusieurs avantages, le principal étant que le mode d'opération est similaire à ce à quoi pourrait ressembler une application en mode d'opération en temps réel. L'application finale est donc dans un état permettant la mise en production avec seulement des modifications mineures. Dans le cas présent cependant, l'agent n'interagit pas directement avec l'environnement, mais propose une classification de l'activité en cours. La Figure 3.5 montre la structure générale de l'application et comment les différentes

parties interagissent entre elles.

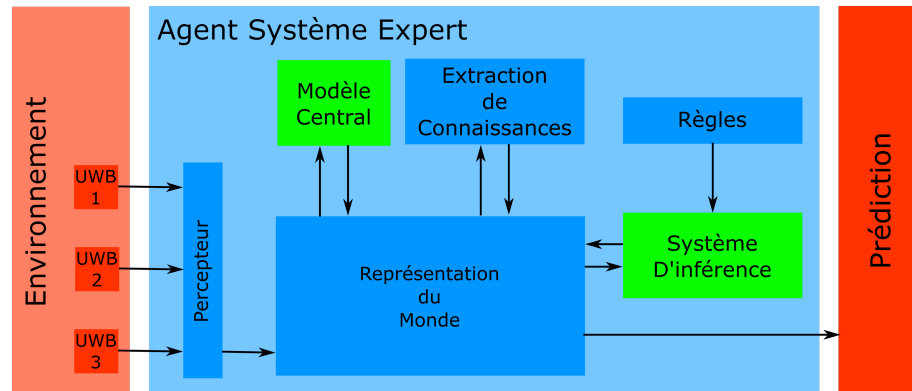


Figure 3.5 : Architecture du système expert.

Lors de l'exécution, les étapes suivantes sont réalisées à chaque coup d'horloge :

1. Les données des radars sont fournies à l'agent par les percepteurs.
2. Les données sont conservées dans une base de données (représentation du monde).
3. Le modèle central est appliqué sur les données les plus récentes et la sortie ajoutée à la base de données.
4. Des connaissances supplémentaires sont dérivées des données brutes fournies par les radars UWB.
5. Le système d'inférence est exécuté : les règles sont appliquées, puis leurs valeurs accumulées dans la base de données.
6. Les activités les plus probables sont ensuite données à la sortie.

En ce qui concerne la programmation du système expert, la librairie *Experta* (Pérez, 2017) a été utilisée. Cette librairie a été identifiée puisqu'elle est une des rares librairies plus

ou moins actives sur les systèmes experts et est basée sur *PyKnow*, une librairie largement référencée.

Un des points importants avec la création d'un système expert est la création de règles. À des fins de simplicité et de respect du concept de système expert, des connaissances sont extraites des données qui permettent de distinguer une activité par rapport à une autre à partir de concepts. Les règles ont été définies de façon instinctive. La principale information exploitée est la position. Avec la position, il est techniquement possible de dire qu'une activité se déroule (ou non) dans un endroit. Par exemple, il est très peu probable que l'activité *Enfiler une veste* est réalisée tandis que la personne se trouve dans la cuisine. De plus, il est également possible de distinguer deux activités en fonction de la distance parcourue dans l'environnement. Par exemple, les activités *Lire* et *Cuisiner* ne devraient pas posséder une distance parcourue similaire. Au final, les informations suivantes ont été utilisées et/ou explorées pour la création des règles :

- **Niveau d'énergie du signal** : L'énergie d'un signal discret est la somme de tous les échantillons élevés au carré. Cela permet de détecter la quantité d'énergie dans un signal (indiquant de façon générale la quantité de mouvement effectué).
- **Zones** : Les zones précises dans l'environnement. Cette information est utilisée en tandem avec la position de la personne dans l'environnement. Les zones sont définies à partir du plan de l'appartement du LIARA et l'historique de position des participants lors de la réalisation d'une activité. Les zones définies sont *Évier de la cuisine*, *Chaise de*

la chambre à coucher, Table, Cuisine, Salon, Entrée, Salle à manger, Soulier (zone où l'activité *Mettre Soulier* est régulièrement effectuée) et *Médicaments* (zone où l'activité *Prendre des médicaments* est régulièrement effectuée).

- **Position** : Position de la personne dans l'environnement. La position est une zone.
- **Compteur de Position** : Compte du nombre de secondes passées dans une zone donnée.
- **Surface Couverte** : Surface totale couverte par la personne dans la fenêtre de temps.
Information tirée de la position de la personne.
- **Déplacement total** : Total des déplacements dans l'environnement. Information tirée de la position de la personne.

Le modèle profond a été entraîné indépendamment du système expert. Lors de l'entraînement, le modèle atteint un taux de reconnaissance d'environ 100 % alors que le taux de reconnaissance sur le jeu de test stagne autour de 35 % à 40 %. La Figure 3.6 montre la progression du taux de reconnaissance sur le jeu de test et d'entraînement. Cette courbe d'apprentissage montre que le modèle souffre de surapprentissage. En d'autres mots, il a mémorisé le jeu d'entraînement. Dans un scénario de surapprentissage, les données présentes dans le jeu d'entraînement sont reconnues alors que les données qui ne sont pas directement présentes dans le jeu d'entraînement sont correctement reconnues dans une proportion beaucoup plus petite. Pour éviter de sélectionner un modèle surentraîné, le modèle qui offre le meilleur taux de reconnaissance sur le jeu de validation est conservé. Comme il est montré à la Figure 3.6, le meilleur taux de reconnaissance est atteint près de la dixième époque. Pour cette raison, le

modèle utilisé pour le développement du système expert a un taux de reconnaissance d'environ 50 % sur le jeu d'entraînement.

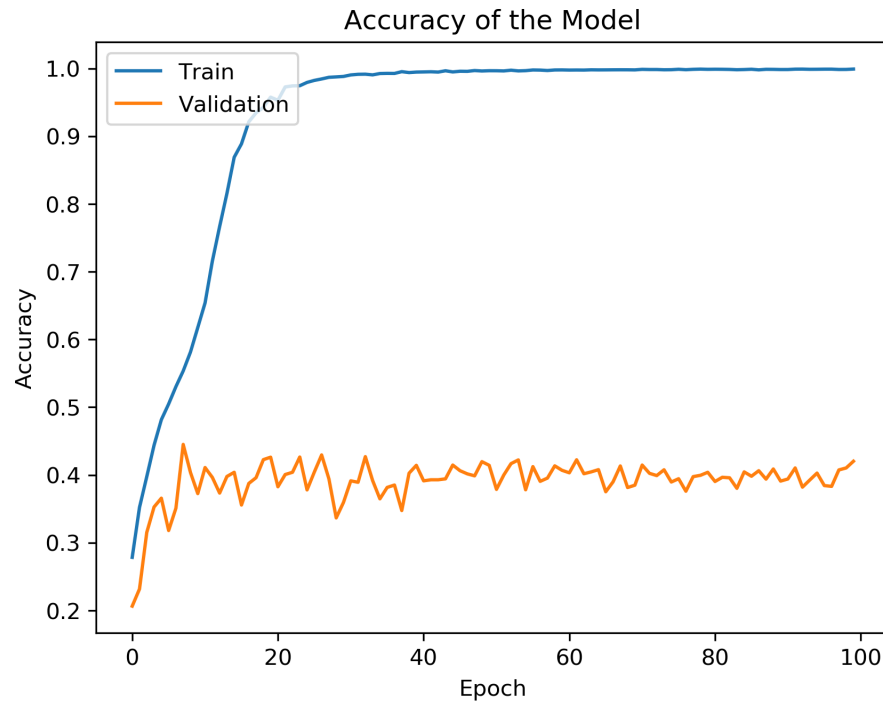


Figure 3.6 : Taux de reconnaissance du modèle *CNN-LSTM* sur le jeu de données LOSO 1.

Pour la classification, plusieurs techniques ont été utilisées pour améliorer la performance de la prédiction. Dans un premier temps, plusieurs activités peuvent être détectées lors d'une inférence (une classification). Pour distinguer quelle activité doit être priorisée, un ordre de priorité pour chaque activité est établi. De plus, lors de la classification d'une activité, une deuxième classification est faite en même temps. Il s'agit ici d'identifier l'autre activité possible en cas d'une mauvaise détection. Par exemple, dans la règle qui reconnaît l'activité *Faire la cuisine*, le deuxième essai est *Faire du thé*. De plus, pour offrir un Top-N de taille constante comme le fait le modèle central, des classifications par défaut sont ajoutées. Les

classifications par défaut peuvent être vues comme les classifications qui ont le plus de chance, a priori, d'être effectuées à un moment donné.

La priorisation des activités a été optimisée avec des algorithmes génétiques (Mitchell, 1998). Pour ce faire, la librairie Python *PyEasyGA* (Remi-Omosowon et Pyeasyga, 2014) est utilisée. Cette librairie permet d'exploiter facilement des algorithmes génétiques. Dans le but d'optimiser l'ordre de priorité des activités, un individu est constitué d'un ordre de priorisation des activités où les indices inférieurs représentent une priorisation plus grande. Ainsi, l'activité à l'indice 0 est considérée comme étant l'activité la plus priorisée par rapport aux autres. Une population de 500 individus générés aléatoirement a été utilisée. L'optimisation s'est déroulée pour une durée de 200 générations. Les techniques suivantes ont été utilisées pour faire évoluer les individus d'une génération à l'autre :

- **Mutation.** Une mutation permet de modifier un seul individu. Dans le cas présent, une mutation est effectuée en appliquant une permutation aléatoire entre deux indices. La probabilité d'une mutation est de 5 %.
- **Croisement.** Un croisement combine l'état de deux parents pour générer deux enfants. Dans le cas présent, un point de croisement (indice) aléatoire est sélectionné. La première partie avant ce point de croisement est conservée dans chacun des enfants. Ainsi, l'enfant A hérite du segment pré point de croisement du parent A et l'enfant B celui du parent B. Le reste des activités dans les enfants est comblé en respectant l'ordonnancement présent dans l'autre parent. Donc, pour chaque indice du parent B, de la plus prioritaire

à la moins prioritaire, si l'activité n'est pas présente dans l'enfant A elle est ajoutée à la fin de la liste de priorité de l'enfant A. La probabilité d'un croisement est de 50 %.

— **Élitisme.** Afin d'éviter que le meilleur individu soit perdu d'une génération à l'autre, ce dernier est ajouté à la prochaine génération sans altération.

Puisque la solution du système expert est conçue pour opérer en temps réel, il est possible de stabiliser la classe détectée à la sortie de la solution en changeant de classe de sortie après un nombre fixe (plus grand que 1) de détections consécutives. La performance du système peut aussi être améliorée légèrement. Le raisonnement derrière cette amélioration est qu'il est très peu probable qu'une personne change successivement d'activités dans un très court laps de temps. Cette approche peut cependant être appliquée dans n'importe quelle autre solution qui effectue la reconnaissance en temps réel.

3.3.3 RÉSULTATS ET DISCUSSION

Les informations les plus évidentes à utiliser pour la détection sont la prédiction du modèle central, qui donne une bonne indication quant à l'activité en cours, et la position de la personne. Il est important encore une fois de préciser que la position estimée de la personne peut-être différente de la position actuelle de la personne dans l'environnement. Cela étant dû à une grande quantité de bruit dans les données, d'angles morts présents dans l'environnement en plus de personnes en mouvement à l'extérieur de l'environnement.

Il est important en premier lieu d'identifier où des gains peuvent être accomplis. Il existe

une différence importante dans la capacité du modèle central à reconnaître des activités courtes (100 secondes et moins, ce qui correspond à la moitié des activités). La plus grande partie des efforts a été mise pour pallier à cette différence. Les expérimentations n'ont pas donné les résultats espérés. Au bout de plusieurs semaines de développements et de tests, tous les efforts d'amélioration ont fini par souffrir de plusieurs problèmes. Dans un premier temps, ce qui est gagné d'un côté est perdu de l'autre. Intuitivement, il est normal que lorsqu'on priorise la détection de la classe A par rapport à la classe B, que les détections correctes de la classe B risquent d'être négativement impactées au profit de fausses détections de la classe A. Cela est encore plus problématique lorsque plusieurs classes sont concernées et qu'un effet domino apparaît. La Figure 3.7 montre le taux de reconnaissance balancé du système expert développé et du modèle central (*CNN-LSTM*) sur le jeu de données d'entraînement LOSO 1.

La Figure 3.8 montre la même chose, mais pour le jeu de données de test.

Dans la Figure 3.7, il est possible de constater que des gains notables ont été effectués sur les activités courtes, cela est également reflété dans la Figure 3.8 pour ce qui attrait au jeu de test. En même temps, les performances sur les activités longues se voient réduites dans les deux cas. Il est aussi possible de constater que les améliorations faites sont seulement présentes dans le Top-1 alors que dans le reste des Top-N, la performance du système expert reste inférieure aux performances du modèle central.

Une des limitations principales des systèmes experts est l'effort requis pour la concep-

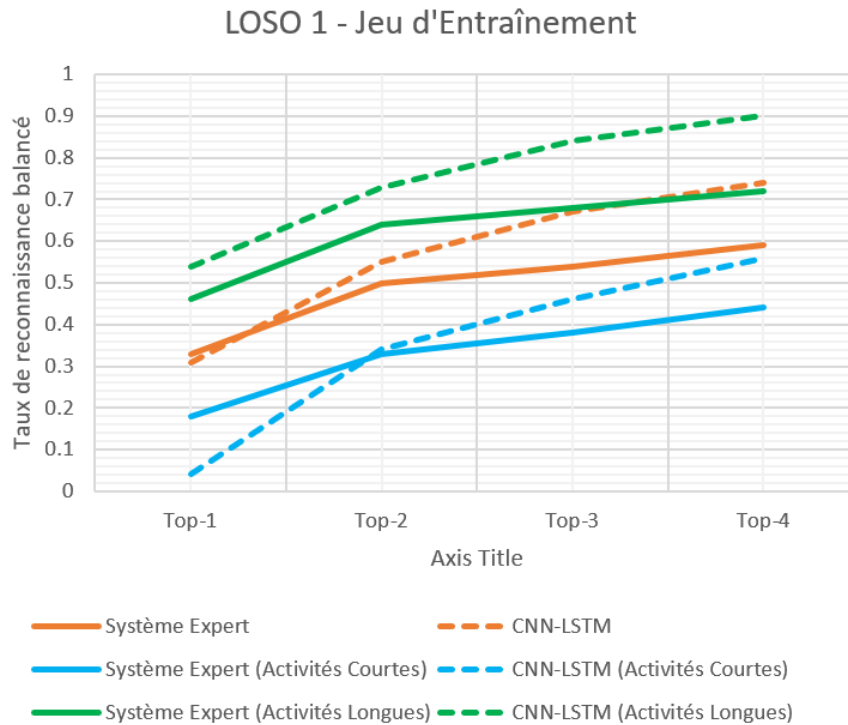


Figure 3.7 : Comparaison du taux de reconnaissance balancé du Top-4 du modèle central (*CNN-LSTM*) et du système expert sur l'ensemble des classes, les classes longues et les classes courtes sur le jeu de données d'entraînement LOSO 1.

tion. Beaucoup d'analyses ont été faites pour pouvoir combler certaines lacunes du modèle précédemment développé au LIARA, et ce, sur un seul des jeux de données LOSO. Ce qui est généralement le cas avec les systèmes experts. Ce qui est problématique cependant, avec l'effort de développement requis, est qu'une solution est fonctionnelle pour un seul jeu de données. La raison fondamentale derrière ce problème est que le modèle n'a pas les mêmes angles morts d'un jeu de données LOSO à l'autre, et il n'est pas garanti que les règles améliorent les performances entre le jeu d'entraînement et le jeu de test. Cela est entre autres observable avec le jeu de données LOSO 1 aux Figures 3.7 et 3.8 où il est possible de constater une différence notable de performance entre le jeu d'entraînement et le jeu de test. Ceci est encore

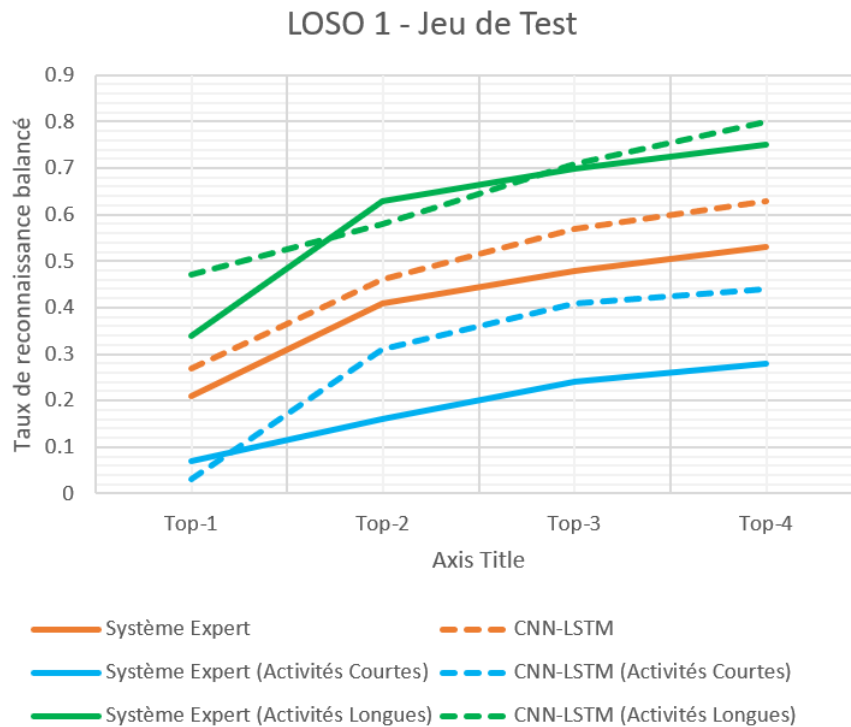


Figure 3.8 : Comparaison du taux de reconnaissance balancé du Top-4 du modèle central (*CNN-LSTM*) et du système expert sur l'ensemble des classes, les classes longues et les classes courtes du jeu de données de test LOSO 1.

plus vrai entre deux jeux de données LOSO. La Figure 3.9 montre le taux de reconnaissance balancé du système expert développé et du modèle central (*CNN-LSTM*) sur le jeu de données d'entraînement LOSO 2. La Figure 3.10 montre la même chose, mais pour le jeu de données de test. Il est possible de constater que les règles développées pour le jeu de données LOSO 1 ne sont pas aussi efficaces pour le jeu de données LOSO 2.

Avant de poursuivre, il est important de se demander pourquoi il serait souhaitable d'utiliser un système expert. L'avantage premier de l'utilisation des systèmes experts est qu'il

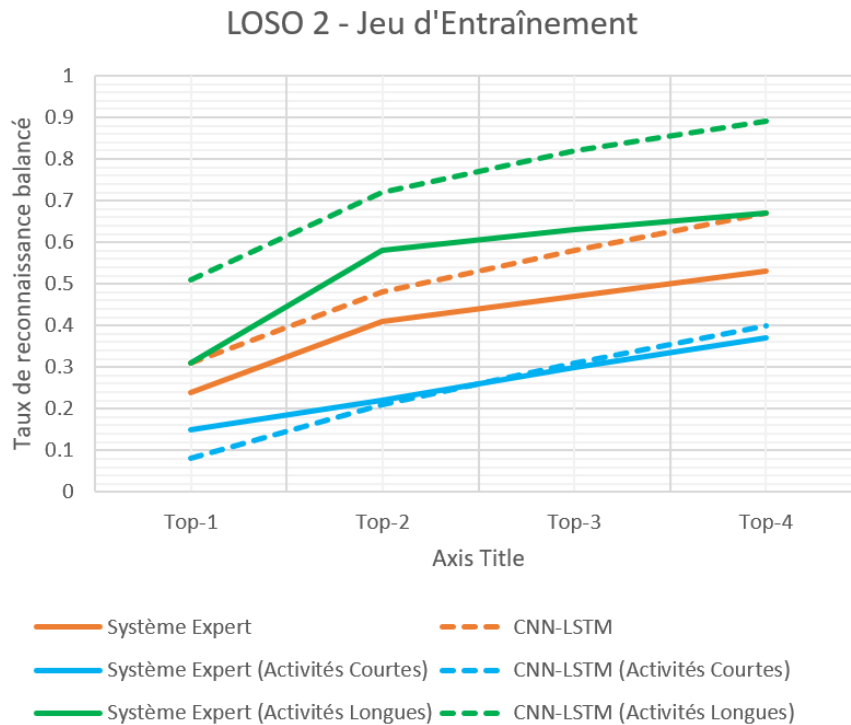


Figure 3.9 : Comparaison du taux de reconnaissance balancé du Top-4 du modèle central (*CNN-LSTM*) et du système expert sur l'ensemble des classes, les classes longues et les classes courtes sur le jeu de données d'entraînement LOSO 2.

est possible de comprendre facilement pourquoi une décision est effectuée par rapport à une autre, car le raisonnement est contenu sous la forme de règles définies par une personne et basé sur des faits. L'interprétabilité du raisonnement d'algorithmes intrinsèquement opaques est une tendance grandissante en intelligence artificielle, et particulièrement dans le domaine de l'apprentissage profond (Doshi-Velez et Kim, 2017). Malheureusement, cet aspect n'est pas dans les objectifs des travaux de ce mémoire. Un autre point important concerne la qualité du jeu de données pour l'extraction de règles. Comme mentionné dans la section précédente, aucune information supplémentaire à celle contenue dans les données brutes des radars n'est donnée. Par exemple, la date et l'heure à laquelle les activités ont eu lieu pourraient très

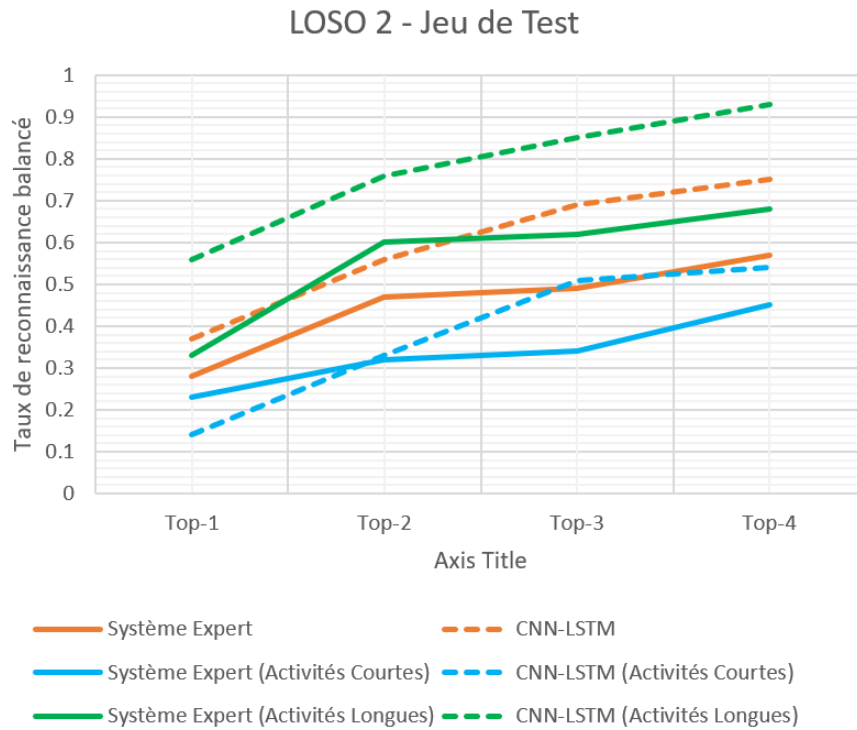


Figure 3.10 : Comparaison du taux de reconnaissance balancé du Top-4 du modèle central (*CNN-LSTM*) et du système expert sur l'ensemble des classes, les classes longues et les classes courtes du jeu de données de test LOSO 2.

bien être utilisées puisque cette information influence les probabilités a priori qu'une action soit réalisée. De plus, la seule information qui est facilement exploitable est la position. La position qui, comme discuté précédemment, est bruitée et comporte quelques angles morts dans l'environnement. Il est donc difficile d'extraire des règles et des faits de façon plus efficace que ne le ferait un classifieur ou un réseau de neurones. Il semble donc que le jeu de données ne soit pas particulièrement idéal pour l'extraction de connaissances intuitives. De plus, rien n'empêche l'utilisation d'un classifieur (réseau de neurones, KNN, SVM ou Forêt d'arbres de décision) ou d'intégrer directement les informations tirées des règles dans le modèle central considérant que toutes les données générées (voir Section 3.3.2) seraient exploitables par ces

classieurs. Puisque le système expert n'est pas utilisé dans le but de justifier le raisonnement derrière la reconnaissance, que les données pourraient facilement être exploitées par d'autres modèles et le temps monumental requis pour créer/ajuster les règles, il est donc discutable d'utiliser un système expert par rapport à d'autres solutions.

Malgré ce qui a été dit, cela n'empêche pas que les réseaux de neurones peuvent apporter une plus-value pour la reconnaissance d'activités. Par exemple, l'ajout du système expert serait très pertinent pour faire du raffinement d'activités. Dans ce contexte, un modèle pourrait être utilisé pour faire la reconnaissance générique d'activités et le système expert pourrait identifier la variation de l'activité avec des informations qui ne devraient pas être pertinentes pour la reconnaissance générale d'une activité. Par exemple, l'activité *Manger* devrait toujours être reconnue comme manger par le modèle de reconnaissance, peu importe quand cela se produit, mais un système expert pourrait, avec une mémoire des activités réalisées et de l'heure durant laquelle l'activité se déroule, déterminer s'il s'agit du déjeuner, du dîner ou du souper. Un autre exemple pourrait déterminer la qualité de la tâche réalisée comme dans le cas où la personne se brosse les dents pas assez longtemps ou trop longtemps. Un exemple final pourrait consister en la classification précise de l'activité marcher en fonction du point de départ et de la destination de la personne.

3.4 DÉVELOPPEMENT D'UN MODÈLE D'APPRENTISSAGE PROFOND

Considérant les travaux réalisés dans la section précédente, il a été décidé d'explorer quels autres types de modèles offriraient une meilleure performance brute pour la reconnais-

sance d'activités. Le chapitre 2 a permis de montrer qu'un grand intérêt réside dans les réseaux de neurones. Cette section vise donc à décrire le développement du modèle CNN construit dans le but de reconnaître les activités enregistrés au LIARA.

3.4.1 TRAITEMENT DE DONNÉES

La possibilité explorée dans ce mémoire est d'entrevoir la reconnaissance d'activités avec des radars UWB comme étant un problème de vision. C'est-à-dire que les données peuvent être transformées et traitées comme l'on ferait avec des images. Pour ce faire, il faut extraire l'activité dans une seule structure qui contient l'information sur la position et les mouvements de la personne dans l'environnement. Puisqu'il y a 3 radars dans l'environnement, il est possible de facilement visualiser la structure combinée en la représentant comme une image RGB où chaque couleur est remplacée par les données d'un radar. Cela permet de déterminer si des caractéristiques sont visuellement présentes dans les données. Les radars opèrent à 50 *scans* par seconde. Puisque 15 secondes de données sont utilisées pour faire une seule détection et que chaque radar retourne 184 *bins* représentant les données Distance-Doppler, la structure combinée des radars a une forme de 750 *scans* par 184 *bins* par 3 radars UWB. Lors de la couche de convolution, la taille du modèle augmente énormément. La plupart des modèles CNN pour la reconnaissance d'objets utilisent des images de tailles plus petites (généralement $180 \times 180 \times 3$) pour réduire la taille des modèles. Pour atteindre une taille plus petite, la structure combinée est compressée sur l'axe temporel (scan) pour atteindre une taille plus facilement exploitable de $150 \times 184 \times 3$. Pour la compression, la structure

est séparée en intervalles de 5 *scans* et le maximum est conservé pour chaque intervalle de distance. À ce point, les structures comprennent les retours des objets dans l'environnement. Afin de soustraire l'environnement et de conserver seulement les mouvements (*Background Extraction*), le premier *scan* de chaque acquisition est soustrait au reste de la structure.

La Figure 3.11 contient deux exemples de données prétraitées. Il est intéressant d'observer qu'une différence visuelle existe entre les activités *Marcher* et *Faire le ménage*. Ce qu'on peut comprendre de ces images, c'est que dans le cas de l'activité *Marcher*, la personne s'approche puis s'éloigne d'un des radars UWB et que dans l'autre cas, elle fait des va-et-vient devant le même radar UWB.

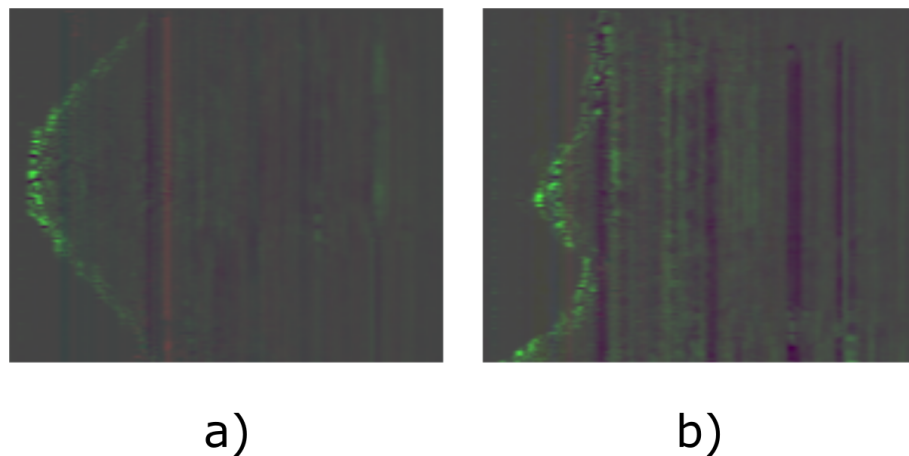


Figure 3.11 : Deux activités prétraitées pour le modèle d'apprentissage profond. a) Activité *Marcher*. b) Activité *Faire le ménage*.

Considérant la quantité de données disponibles, un grand chevauchement existe entre deux échantillons extraits consécutivement lors de la création du jeu de données. Afin d'éviter que des données trop similaires se retrouvent simultanément dans le jeu d'entraînement du

modèle et dans le jeu de test du modèle, la validation croisée LOSO est utilisée. De cette façon, 10 jeux de données différents peuvent être créés, soit un par participant.

3.4.2 ARCHITECTURE PROPOSÉE

La première étape de la création d'un modèle pour la reconnaissance consiste en la comparaison de différentes architectures déjà présentes dans la librairie *Keras*⁸. Cette étape permet de vérifier qu'il est possible d'utiliser les données telles que prétraitées, puis d'identifier quelle architecture est prédisposée à mieux performer sur le jeu de données. Les modèles suivants ont été testés avec le jeu de données LOSO 1 : VGG16 (Simonyan et Zisserman, 2015), VGG19 (Simonyan et Zisserman, 2015), Xception (Chollet, 2016) et *EfficientNet* B0 à B6 (Tan et Le, 2019). De plus, puisque les autoencodeurs ont montré un certain succès dans ce genre de scénario (voir Sadreazami *et al.* (2020)), un modèle autoencodeur a été développé et également testé. L'architecture complète de l'autoencodeur développé est montrée dans le Tableau 3.2. En ce qui concerne la signification des couches convolutives, celles-ci sont décrites dans la Section 2.1.2. Les autres couches non couvertes par cette section sont décrites ci-dessous ;

— **Dropout.** La couche de type *dropout* sert à régulariser l'apprentissage du réseau de neurones en laissant tomber aléatoirement un certain nombre de neurones dans la couche.

Cela permet d'éviter que le réseau de neurones surapprenne et se fie seulement à la

8. <https://keras.io/api/applications/>

valeur d'un nombre restreint de neurones.

- **Flatten**. La couche *flatten* permet de prendre la sortie multidimensionnelle d'une couche et de la transformer en sortie unidimensionnelle.
- **Dense**. Nom communément donné aux couches entièrement connectées.

ID	Nom de la couche	Forme de sortie
1	Input	$150 \times 184 \times 3$
2	Zero_Padding	$152 \times 184 \times 3$
3	Convolution_1	$152 \times 184 \times 32$
4	Max_Pooling_1	$76 \times 92 \times 32$
5	Convolution_2	$76 \times 92 \times 64$
6	Max_Pooling_2	$38 \times 46 \times 64$
7	Convolution_3	$38 \times 46 \times 128$
8	Flatten	223744
9	Dense_1	200
10	Dropout_1	200
11	Dense_2	50
12	Dropout_2	50
13	Output	15

Tableau 3.2 : Architecture de l'autoencodeur développé.

Le résultat de l'expérimentation est montré au Tableau 3.3. Pour cette expérimentation, le résultat est calculé une seule fois, et sur un seul jeu de données LOSO. La raison étant que le temps requis pour faire un seul entraînement avec le matériel utilisé qui est long. De plus, le but ici est d'avoir une idée générale de la performance du modèle afin de se concentrer

sur celui qui offre les meilleures performances et non de mesurer la performance exacte de chaque modèle. Il est possible de constater que le *EfficientNet* (peu importe la version, mais principalement B0) offre les meilleures performances, suivi de l'autoencodeur et de Xception. *EfficientNetB0* est le modèle de base (donc plus petit) proposé par Tan et Le (2019). La raison du gain de performance de *EfficientNetB0* par rapport aux autres versions tient probablement du fait que le jeu de données est limité, et que des architectures plus larges et plus profondes sont plus sujettes au surapprentissage. Il est donc intéressant ici d'identifier que l'exploration de l'utilisation des autres versions d'*EfficientNet* serait importante si un jeu de données plus complet était créé dans le futur.

Modèle	Taux de reconnaissance
VGG16	0,081
VGG19	0,081
Xception	0,546
EfficientNetB0	0,626
EfficientNetB4	0,612
EfficientNetB5	0,623
EfficientNetB6	0,601
Autoencoder CNN	0,554

Tableau 3.3 : Performance de différents modèles sur le jeu de données développé à partir de scans de radars UWB.

Une expérimentation supplémentaire a été réalisée sur le modèle *EfficientNetB0* concer-

nant la taille des filtres utilisés. Normalement, les filtres dans un CNN sont carrés. La raison étant que les filtres permettent d'extraire des modèles servant à identifier des objets dont on ne connaît ni la forme ni l'orientation des objets à détecter. Or, dans le cas présent, nous avons une intuition sur la forme des modèles. Il est raisonnable de penser que les modèles recherchés sont quant à eux orientés dans le temps. Pour valider l'intérêt de changer la forme des filtres, plusieurs combinaisons de tailles ont été testées. La performance des filtres a été testée sur un seul jeu de données LOSO, et sur un seul entraînement. La raison étant le temps requis pour tester chaque configuration à plusieurs reprises. L'entièreté des jeux de données LOSO est chronophage et pourrait prendre plusieurs centaines d'heures de calcul avec le matériel informatique utilisé (Intel i5-8600k et NVIDIA GTX 1080). Le Tableau 3.4 contient les différents tests de performance de différentes tailles de filtres. Dans ce tableau, la taille des filtres est assignée par bloc architectural du réseau de neurones. L'architecture d'*EfficientNet*, comme plusieurs autres CNN, est divisée en plusieurs blocs. Les couches au sein d'un même bloc partagent les mêmes caractéristiques. Cette approche est pratique lorsque les réseaux de neurones atteignent une certaine profondeur et qu'il devient peu pratique d'ajuster séparément chaque couche. Considérant le temps requis pour l'entraînement, chaque configuration de filtres a été testée une seule fois avec le jeu de données LOSO 1. La meilleure augmentation de performance est donnée à l'essai 2. L'essai 2 offre une augmentation de 5,5 % par rapport au modèle d'origine. Puisque le nombre de combinaisons possibles est important et que le temps requis pour entraîner le modèle sur un jeu de données LOSO est long, une recherche exhaustive de la meilleure combinaison n'a pas été faite. Cette expérimentation a tout de

même permis une meilleure performance du réseau de neurones et laisse présager qu’une autre combinaison de tailles de filtres offrant de meilleures performances existe. À des fins de clarté, le modèle avec taille de filtres modifiée sera appelé *Tuned EfficientNetB0* pour le reste du mémoire.

Essai	Bloc 1	Bloc 2	Bloc 3	Bloc 4	Bloc 5	Bloc 6	Bloc 7	Taux de reconnaissance	Taux de reconnaissance balancé
Point de départ	3×3	3×3	5×5	3×3	5×5	5×5	3×3	0,626	0,644
1	2×5	2×5	3×8	2×5	3×8	3×8	3×8	0,612	0,628
2	3×3	4×4	3×8	2×5	2×8	3×8	5×10	0,678	0,699
3	3×3	4×4	4×6	3×6	3×8	4×8	5×10	0,657	0,667
4	6×6	6×6	4×6	3×6	3×8	3×8	3×10	0,617	0,64
5	3×3	5×5	3×5	2×6	2×8	2×8	2×10	0,595	0,616
6	3×3	3×3	3×4	3×5	3×6	3×8	5×10	0,61	0,631
7	3×3	3×3	3×8	3×8	3×8	3×8	5×10	0,606	0,627
8	3×3	4×4	2×4	2×5	3×8	3×8	5×10	0,584	0,604
9	3×3	4×4	4×8	2×6	2×8	3×8	2×5	0,66	0,675

Tableau 3.4 : Variations de la taille des filtres et leur performance respectives.

Comme discuté à la Section 3.2, la position de la personne dans l’environnement est constamment calculée. Afin de fournir des informations supplémentaires au modèle final, la position est, quant à elle, analysée par un autre réseau de neurones de type LSTM. Le modèle LSTM est explicité dans le Tableau 3.6. Le fonctionnement des couches LSTM est expliqué à

la Section 2.1.2. Il est important de mentionner encore une fois que les résultats ne sont pas meilleurs que ceux obtenus avec les autres modèles testés. Puisque le calcul de la position de la personne dans l'environnement est basé sur des données bruitées, la classification de l'activité en utilisant seulement cette information est fortement impactée (négativement) par ce bruit.

ID	Nom de la couche	Taille de sortie
1	Input	15, 5
2	LSTM_1	15, 50
3	LSTM_2	50
4	Dense_1	100
5	Dropout_1	100
6	Dense_2	100
7	Dropout_2	100
8	Dense_3	50
9	Dropout_3	50
10	Output	15

Tableau 3.5 : Architecture du modèle *LSTM*.

Le modèle final, tel qu'illustré à la Figure 3.12, est donc composé du *LSTM* et de *Tuned EfficientNetB0*. Considérant la taille du modèle final, l'entraînement est fait en trois temps. Dans un premier temps, *Tuned EfficientNetB0* est entraîné sur le jeu de données prétraitées. Ensuite, le *LSTM* est entraîné sur les données de position. Finalement, les poids des couches des deux modèles sont transférés (apprentissage par transfert) sur le modèle final. Les poids de ces couches fraîchement transférés sont figés pour les empêcher de changer lors de la

première partie de l'apprentissage qui vise seulement à faire apprendre les dernières couches denses, responsables de la classification. Lors de la deuxième partie de l'apprentissage, toutes les couches sont défigées afin d'optimiser le modèle de bout en bout. Pour l'apprentissage, l'optimiseur *Adam* (Kingma et Ba, 2017) est utilisé avec un taux d'apprentissage de 0,001.

L'entraînement se fait sur un nombre relativement petit d'époques. La taille relativement petite du jeu de données en est la cause, puisque les données sont apprises rapidement par coeur par le modèle. Les meilleurs résultats sur *Tuned EfficientNetB0* et le *LSTM* sont atteints avant la marque des 25 époques. L'architecture combinée de *Tuned EfficientNetB0* avec *LSTM*, quant à elle, atteint un pic avant la marque des 10 époques. Lors de l'entraînement, le modèle qui a un meilleur taux de reconnaissance balancé sur le jeu de test est conservé.

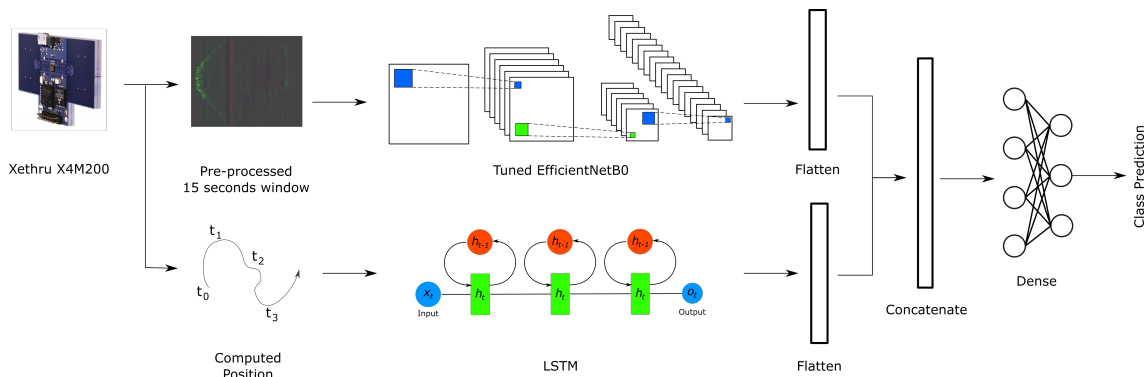


Figure 3.12 : Architecture du modèle final.

3.4.3 RÉSULTATS ET DISCUSSION

Deux métriques principales ont été utilisées pour établir la performance du modèle ; le taux de reconnaissance et le taux de reconnaissance balancé. Le taux de reconnaissance permet

de mesurer la capacité d'un modèle à faire la prédiction correcte d'une classe et est calculé en divisant le nombre de prédictions correctes par le nombre total de prédictions effectuées. Dans ce cas, on veut avoir un modèle qui tend vers un taux de reconnaissance de 1 (100 %).

Dans le cas présent, les classes n'ont pas le même nombre d'instances. L'utilisation du taux de reconnaissance est donc plus problématique puisque les inaptitudes du modèle à classer les instances de classes moins nombreuses sont cachées par la performance sur les plus grandes classes. Un exemple extrême de ce problème sur un classifieur à deux classes pourrait ressembler à ceci : la classe A contient 1000 instances, dont 1000 instances ont été correctement classées par un classifieur alors que la classe B qui contient une seule instance qui n'a pas été correctement classée. Dans ce scénario, le taux de reconnaissance du modèle est de 99,9 %, alors que le classifieur est seulement capable de classer une seule classe sur deux. Une alternative facile au taux de reconnaissance est le taux de reconnaissance balancé. Avec le taux de reconnaissance balancé tel que décrit à l'équation (3.1) où n est le nombre de classes, VP_i est le nombre de vrais positifs pour une classe i et FN_i le nombre de faux négatifs pour une classe i , le taux de reconnaissance de chaque classe est considéré à parts égales. Pour revenir sur l'exemple précédent, le taux de reconnaissance balancé serait de 50 %, ce qui représente beaucoup mieux l'incapacité du classifieur à classer une classe sur deux.

$$\frac{1}{n} \sum_{i=1}^n \frac{VP_i}{VP_i + FN_i} \quad (3.1)$$

La Figure 3.13 montre le taux de reconnaissance moyen des modèles sur l'ensemble des

jeux de données LOSO. Cette figure montre que *Tuned EfficientNetB0 avec LSTM* offre un meilleur taux de reconnaissance que les autres modèles, montrant une augmentation de 18,63 % par rapport au modèle proposé par les travaux antérieurs (Maitre *et al.*, 2021). L'augmentation est d'autant plus grande lorsque l'on compare le taux de reconnaissance balancé moyen sur l'ensemble des jeux de données LOSO tel que montré à la Figure 3.14. Il est possible de constater une augmentation de 29,08 % sur le taux de reconnaissance balancé du Top-1. Cela montre que le modèle développé est meilleur que le modèle de base pour classer les différentes classes.

Un autre aspect intéressant à comparer est la stabilité des modèles. Un modèle est considéré comme stable sur un jeu de données de type LOSO si le taux de reconnaissance du modèle varie peu d'un jeu de données LOSO à un autre. La Figure 3.15 montre l'écart type du taux de reconnaissance des différents modèles sur l'ensemble des jeux de données LOSO. Dans un même ordre d'idées, la Figure 3.16 montre l'écart type du taux de reconnaissance balancé des différents modèles sur l'ensemble des jeux de données LOSO. Il est possible de constater dans les deux cas que la sortie du modèle *Tuned EfficientNetB0 avec LSTM* est plus stable que le modèle de base. Il s'agit ici d'une autre amélioration (en plus du taux de reconnaissance) qu'offre le modèle développé.

La Figure A.1, située à l'Annexe A, contient la matrice de confusion du Top-1 du modèle *CNN-LSTM* sur le jeu de données LOSO 1. Dans le même ordre d'idée. La Figure

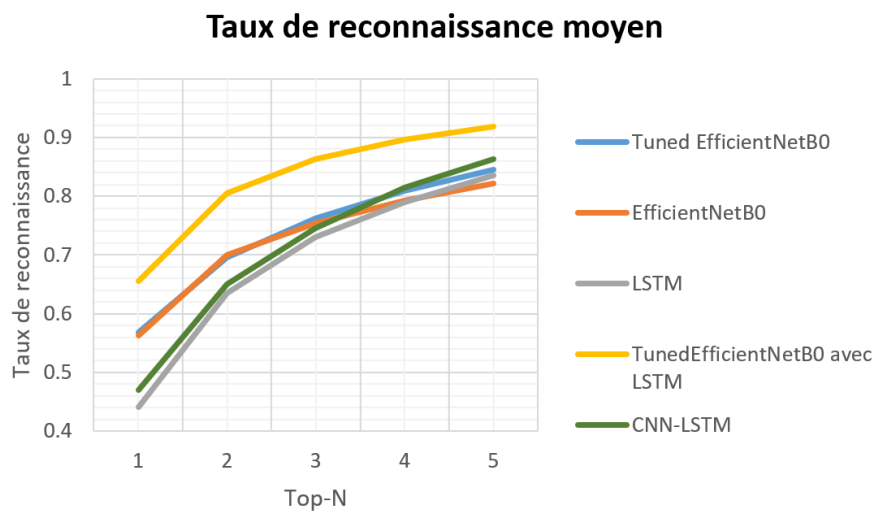


Figure 3.13 : Comparaison du taux de reconnaissance moyen des modèles sur l'ensemble des jeux de données LOSO.

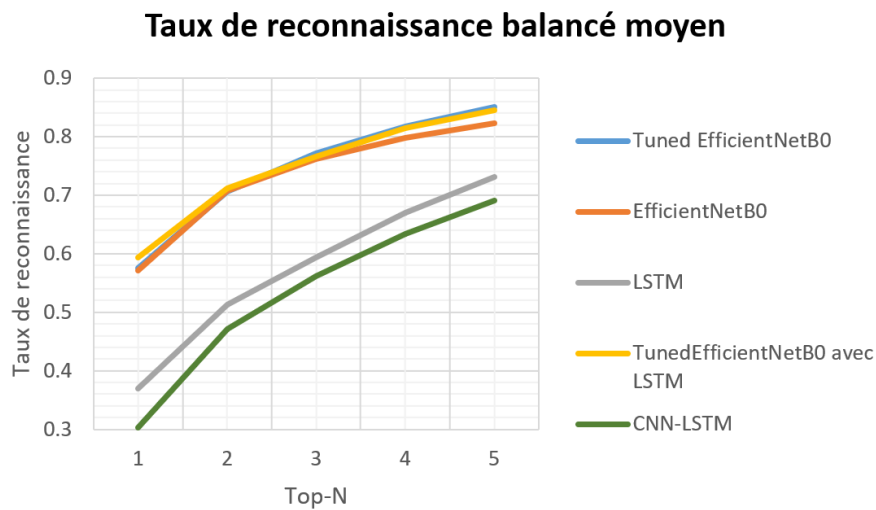


Figure 3.14 : Comparaison du taux de reconnaissance balancé moyen des modèles sur l'ensemble des jeux de données LOSO.

Écart type du taux de reconnaissance

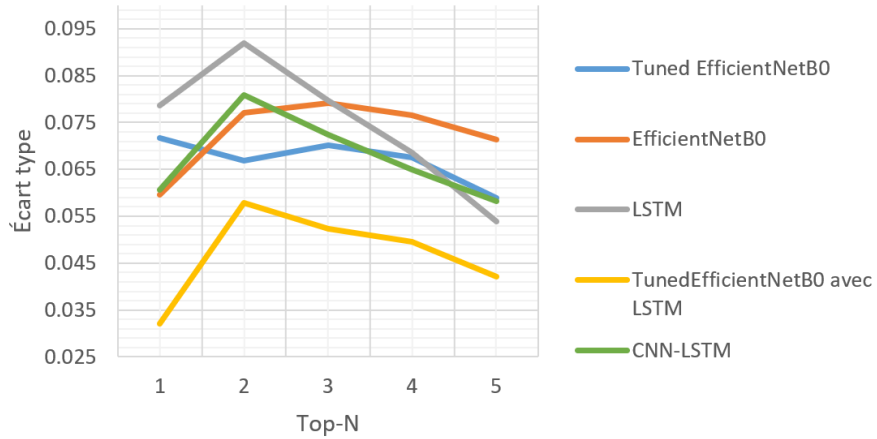


Figure 3.15 : Comparaison de l'écart type du taux de reconnaissance des modèles sur l'ensemble des jeux de données LOSO.

Écart type du taux de reconnaissance balancé

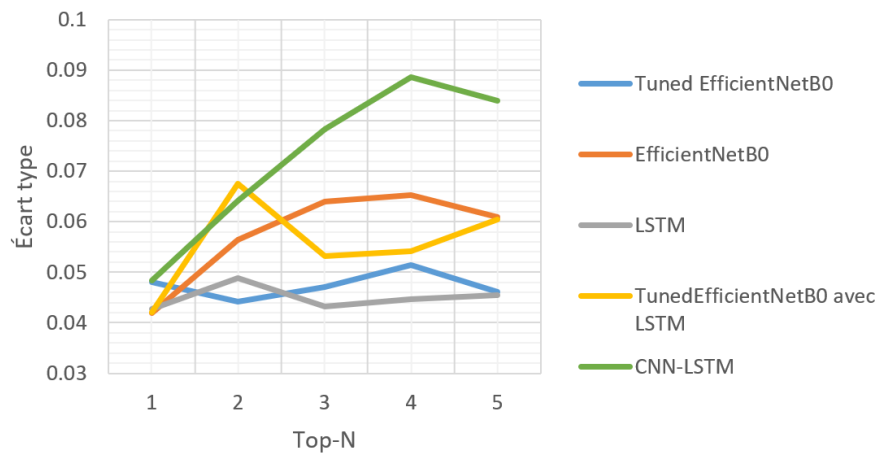


Figure 3.16 : Comparaison de l'écart type du taux de reconnaissance balancé des modèles sur l'ensemble des jeux de données LOSO.

A.2, également située à l'Annexe A, contient la matrice de confusion du Top-1 pour le modèle *Tuned EfficientNetB0 avec LSTM* sur le jeu de données LOSO 1. De ces matrices, des informations intéressantes peuvent être déduites. Premièrement, il est possible de comparer les améliorations sur les activités dites courtes (effectuées en moyenne en moins de 100 secondes, soit les activités *Boire*, *Enfiler une veste*, *Mettre des souliers*, *Dormir*, *Prendre des médicaments*, *Marcher* et *Laver les mains*) et les activités dites longues (soit les activités *Laver les dents*, *Faire le ménage*, *Faire à manger*, *Faire la vaisselle*, *Manger*, *Faire du thé*, *Lire* et *Utiliser un ordinateur*). Ces résultats sont compilés dans au Tableau 3.1. À partir de ces données, il est possible de constater que le gain de performance le plus notable du modèle *Tuned EfficientNetB0 avec LSTM* par rapport au modèle *CNN-LSTM* est causé par la capacité du modèle à reconnaître les activités courtes. Il s'agit d'une augmentation de 51,23 %.

Il est intéressant de constater que malgré une amélioration notable des performances, certaines activités sont tout de même ignorées. Ces activités (*Boire*, *Prendre des médicaments* et *Laver les mains*) ne sont dans les deux cas aucunement reconnues par le modèle. Ces activités ont certains points en commun. Premièrement, peu de mouvement est requis pour effectuer chacune de ces tâches et, deuxièmement, ces événements peuvent être effectués dans la salle de bains, un angle mort potentiel des radars UWB dans l'environnement.

Modèle	Type d'activités	Taux de reconnaissance (%)	Taux de reconnaissance balancé
CNN-LSTM	Toutes	44,51	27,32
Tuned EfficientNetB0 avec LSTM	Toutes	70,39	62,47
CNN-LSTM	Courtes	7,77	5,12
Tuned EfficientNetB0 avec LSTM	Courtes	58,0	50,7
CNN-LSTM	Longues	49,51	46,75
Tuned EfficientNetB0 avec LSTM	Longues	72,64	72,77

Tableau 3.6 : Comparaison de la performance des modèles sur des sous-ensembles du jeu de données LOSO 1.

3.5 ANALYSE DE LA CAPACITÉ DES RADARS UWB POUR LA CAPTATION DE MOUVEMENTS AU LIARA

Comme il a été mentionné dans la section précédente, certaines activités sont difficiles à distinguer. Cette section vise donc à déterminer la source de cette difficulté à reconnaître ces activités.

3.5.1 NOUVEAU JEU DE DONNÉES

Le but de la création de ce jeu de données est multiple. D'abord, on veut vérifier si les radars UWB, la méthode de prétraitement des données et le modèle sont adéquats pour la détection de mouvements variablement grands et petits. Ensuite, on veut mesurer l'impact

que peut potentiellement avoir la hauteur d'installation des capteurs sur la capacité à capter les mouvements. Enfin, on veut évaluer si les angles morts du modèle sur certaines activités (*Boire, Prendre des médicaments et Laver les mains*) sont attribuables à l'endroit où se déroule l'activité.

Pour la capture de mouvements, 10 emplacements ont été identifiés dans l'environnement intelligent du LIARA et sont montrés à la Figure 3.17.

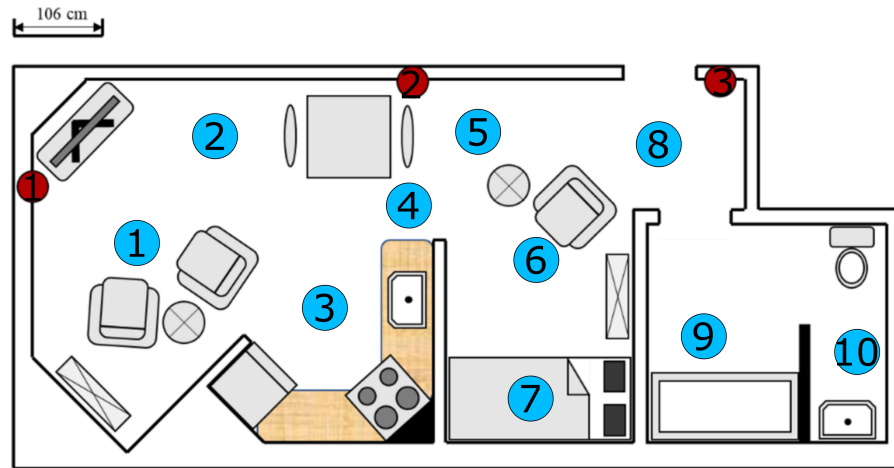


Figure 3.17 : Emplacements prédéterminés pour la création du nouveau jeu de données au LIARA.

Dans cette figure, les 10 emplacements sont numérotés et indiqués par un cercle bleu. La flèche sur chacun des emplacements indique l'orientation de la personne dans l'environnement lors de l'enregistrement. Ces orientations ont été choisies, car elles semblent naturelles par rapport à l'environnement. La position des 3 radars UWB dans l'environnement est quant à elle identifiée en rouge. Ces emplacements ont été sélectionnés pour couvrir l'entièreté de l'environnement. Quatre mouvements ont été identifiés pour la création du jeu de données. Ces

mouvements représentent une gradation de mouvements des bras. Lors de la réalisation des mouvements, la position de la personne est fixe (les pieds ne bougent pas) et les mouvements sont effectués avec les bras pour une durée de 30 secondes. Les mouvements identifiés sont les suivants (à des fins de clarté, des pictogrammes ont été faits pour montrer la posture et la portée des mouvements et sont montrés à la Figure 3.18) :

1. **Aucun mouvement.** Les bras et avant-bras sont placés sur le long du corps.
2. **Mouvements fins.** Les bras sont collés sur le long du corps, les avant-bras placés devant la personne. Les mouvements sont contraints aux poignets et aux mains (doigts).
3. **Mouvements intermédiaires.** Les bras sont collés sur le long du corps. Les mouvements sont contraints aux avant-bras, poignets et mains.
4. **Mouvements grands.** Les mouvements sont contraints aux bras, avant-bras, poignets et mains.

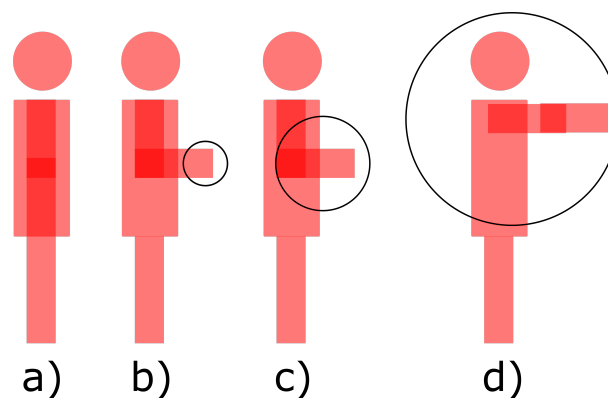


Figure 3.18 : Différents mouvements planifiés pour la collecte de données. a) Aucun mouvement. b) Mouvements fins. c) Mouvements intermédiaires. d) Mouvements grands.

Les 4 activités ont été enregistrées aux 10 positions. Puisque la salle de bains est la seule pièce à avoir une porte, les enregistrements faits aux positions 9 et 10 ont été effectués deux fois, une fois avec la porte ouverte et l'autre avec la porte fermée.

La prise de données a été réalisée deux fois, avec les radars UWB placés à deux hauteurs clés. Dans un premier temps, les capteurs sont placés à 36 cm du sol. Il s'agit de la hauteur utilisée pour l'autre jeu de données utilisé dans ce mémoire. Cette hauteur correspond également à la hauteur à laquelle les prises électriques sont généralement placées. Dans un deuxième temps, les radars UWB sont placés à 96 cm du sol. Cette hauteur concorde avec le centre de gravité de la personne qui a effectué les activités. Cette hauteur a été choisie puisqu'il s'agit de la hauteur approximative où les mouvements fins sont effectués.

3.5.2 VISUALISATION DES DONNÉES

Pour commencer, les données brutes (sans traitement) ont été qualitativement analysées. Afin de comparer la visibilité des différents mouvements, les données de radars UWB sans traitement de chacune des activités enregistrées à un seul endroit sont comparées. La Figure 3.19 contient les quatre activités (*Aucun mouvement*, *Mouvements fins*, *Mouvements intermédiaires* et *Mouvements grands*) enregistrées à la position 1 avec les capteurs installés à 36 cm du sol. À des fins d'affichage, les données sont normalisées. Visuellement, il est très difficile d'identifier des caractéristiques lorsque toutes les données de radars UWB sont présentes. Cependant, lorsque l'on observe seulement les données provenant du capteur le plus proche de la position 1 tel que montré à la Figure 3.20, il est possible de constater que les mouvements

sont visibles et qu'une gradation est visible d'une activité à l'autre. Il est aussi intéressant de noter que la plus petite différence visuelle existe entre l'activité *Mouvements intermédiaires* et *Mouvements grands* et *Aucun mouvement* et *Mouvements fins*.

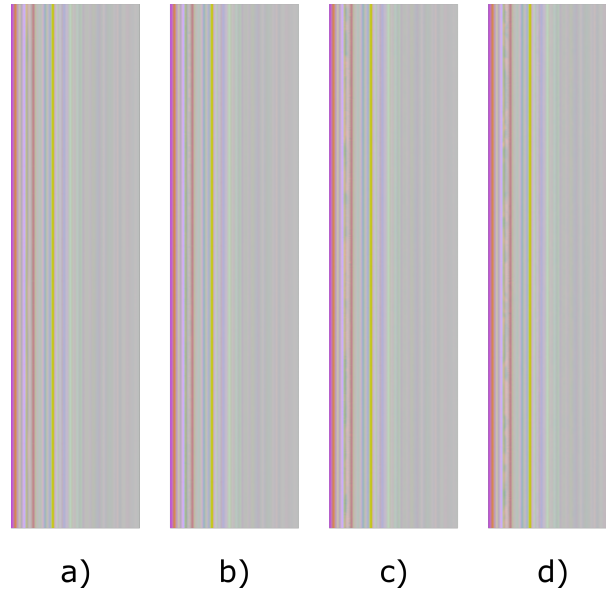


Figure 3.19 : Comparaison des données brutes des mouvements enregistrés à la position 1 avec capteurs installés à 36 cm du sol. a) *Aucun mouvement* b) *Mouvements fins* c) *Mouvements intermédiaires* d) *Mouvements grands*

Dans le but de comparer l'impact de la position sur une activité, la Figure 3.21 contient les données brutes de l'activité *Mouvements grands* aux positions 2, 4 et 9. L'activité *Mouvements grands* a été sélectionnée pour cette comparaison puisqu'elle contient la plus grande quantité de mouvement, et est donc plus susceptible d'être visible dans les données brutes. Les positions 2, 4 et 9 ont été sélectionnées pour cette comparaison puisque celles-ci sont respectivement positionnées près des différents radars UWB et leur font face. La position 2 est proche du radar UWB 1 et lui fait face, la position 4 est proche du radar UWB 2 et lui fait

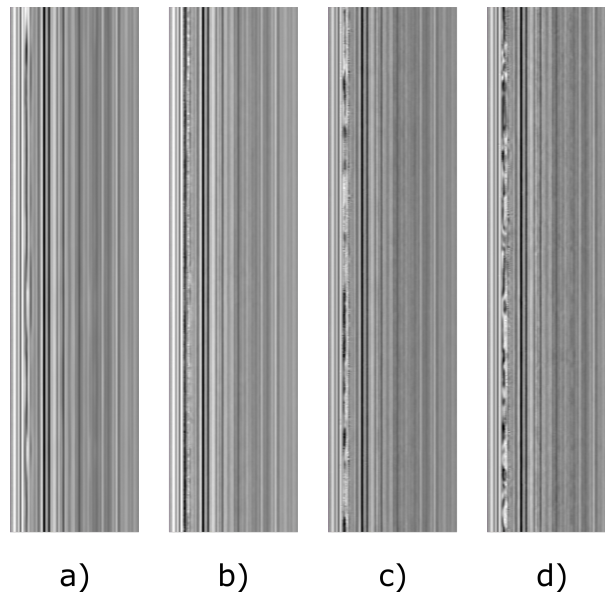


Figure 3.20 : Comparaison des données brutes du radar UWB 1 pour l'ensemble des mouvements à la position 1 avec capteur installé à 36 cm du sol. a) *Aucun mouvement* b) *Mouvements fins* c) *Mouvements intermédiaires* d) *Mouvements grands*

face, et la position 3 (avec porte ouverte) est proche du radar UWB 3 et lui fait face. Le radar UWB le plus près de la position 2 est associé au canal de couleur rouge, le radar UWB le plus près de la position 4 est associé au canal de couleur verte et le radar UWB le plus près de la position 9 est associé au canal de couleur bleu. Il est possible de distinguer les mouvements, principalement à la position 4. Dans le cas de la position 2 et 9, les mouvements sont présents, mais très difficilement visibles, car beaucoup de bruit est présent.

Dans le but de visualiser l'impact du prétraitement de données sur la mise en lumière de patterns, les données prétraitées sont comparées. La Figure 3.22 contient les données traitées des quatre activités (*Aucun mouvement*, *Mouvements fins*, *Mouvements intermédiaires* et *Mouvements grands*) enregistrées à la position 1 avec les capteurs installés à 36 cm du sol. Il

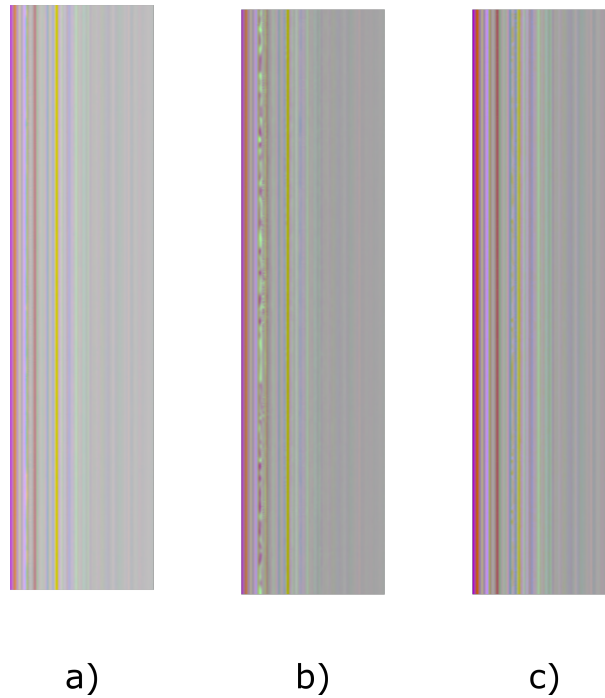


Figure 3.21 : Comparaison des données brutes de l'activité *Mouvements grands* aux positions 2, 4 et 9 avec capteurs installés à 36 cm du sol. a) Position 2 b) Position 4 c) Position 9

est possible de constater que le canal de couleur correspondant au radar UWB le plus près de la position 1 (canal de couleur rouge) prend de plus en plus d'importance dans les données et que le signal semble de plus en plus net. Ainsi, une gradation existe dans les données, reflétant l'augmentation de la quantité de mouvement. La gradation n'est pas aussi flagrante que dans la Figure 3.20. Ceci est causé par la normalisation des données qui est faite une fois le premier *scan* soustrait du reste des données, qui accentue les petits mouvements. Il est donc possible de constater que la méthode de prétraitement permet d'extraire le mouvement des bras des données brutes.

Il est aussi important de vérifier si un *pattern* visuel existe entre les instances d'une

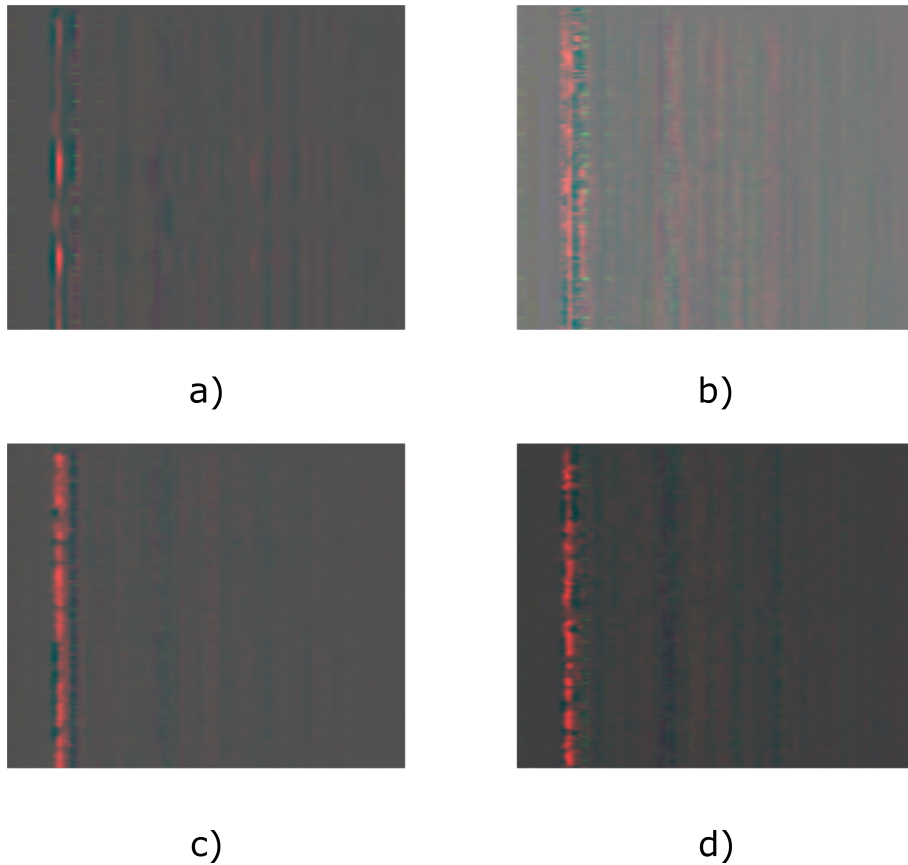


Figure 3.22 : Comparaison des données traitées des activités à la position 1 avec capteurs installés à 36 cm du sol. a) *Aucun mouvement* b) *Mouvements fins* c) *Mouvements intermédiaires* d) *Mouvements grands*

même activité. La Figure 3.23 contient les données prétraitées pour l'activité *Mouvements grands* aux positions 2, 4 et 9 avec les capteurs installés à 36 cm du sol. Il est possible de constater qu'un seul des radars UWB est mis en évidence à chacune des positions. Cela montre que la méthode de prétraitement permet d'extraire le mouvement des bras peu importe la position de la personne dans l'environnement.

Considérant que la méthode de prétraitement permet d'extraire clairement les mouve-

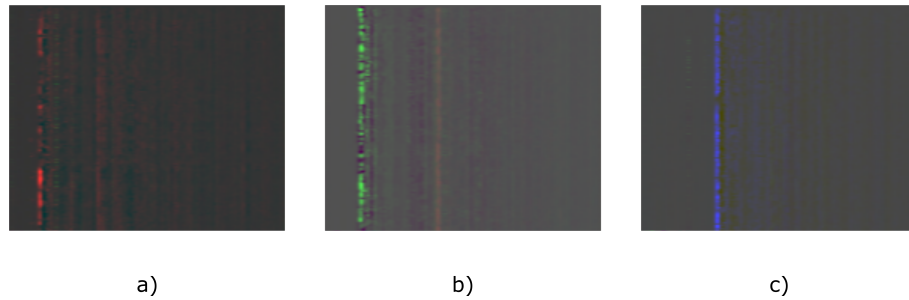


Figure 3.23 : Comparaison des données traitées de l'activité *Mouvements grands* aux positions 2, 4 et 9 avec capteurs installés à 36 cm du sol. a) Position 2 b) Position 4 c) Position 9

ments près de chaque radar UWB, et qu'il est possible de voir une différence entre les instances de différentes activités à une position donnée, il semble que, qualitativement, la méthode de prétraitement des données soit adéquate pour la reconnaissance de mouvements de bras. Une faiblesse potentielle de la méthode de prétraitement est que la normalisation est faite après avoir soustrait le premier *scan* au reste des *scan*. Cela pourrait causer des petits mouvements à sembler plus grands qu'ils le sont réellement.

3.5.3 ANALYSE DE LA PERFORMANCE DU MODÈLE SUR LE NOUVEAU JEU DE DONNÉES

Afin de déterminer si l'architecture telle que développée dans la Section 3.4 est apte à reconnaître les différents types de mouvements, seulement le modèle *Tuned EfficientNetB0* sans la partie *LSTM* a été utilisé. La raison principale pour ce choix est que la partie *LSTM* exploite les données de position calculées. Cette information n'apporte aucune intuition par rapport à l'activité en cours puisque la position de la personne est statique. Dans le meilleur des cas, aucune amélioration ne serait apportée, dans le pire, le bruit de la position pourrait

nuire à la reconnaissance des mouvements.

Les données sont traitées de la même façon que pour le traitement des données dans la Section 3.4.1. Pour chaque activité enregistrée, 30 secondes sont exploitables. Puisque les activités sont très répétitives, les instances extraites des enregistrements sont espacées de 1 seconde, pour un total de 15 instances par enregistrement. Le jeu de données contient donc 720 instances au total et 60 par position. La raison de cette approche est que le nombre d'enregistrements est restreint. Le jeu de données est divisé en 12 sous-jeux de données LOSO, soit un sous-jeu de données par position. Puisque le temps de calcul le permet avec le matériel informatique utilisé, chaque apprentissage a été réalisé 5 fois. Entraîner le modèle à plusieurs reprises permet de déterminer si le modèle est stable et trouve une solution qui est consistante.

Deux variations de l'entraînement ont été réalisées. Dans un premier temps, le modèle a été entraîné à partir de zéro. Pour ce faire, le modèle est initialisé avec des poids aléatoires et l'entraînement est fait sur une période de 100 époques. L'optimiseur *Adam* est utilisé avec un taux d'apprentissage de 0,001.

Dans un deuxième temps, de l'apprentissage par transfert a été fait. L'apprentissage par transfert s'est fait en deux étapes. Pour la première étape, l'apprentissage par transfert est fait sur les couches inférieures du modèle (toutes les couches à l'exception des couches denses responsables de la classification). Lors de cette première partie de l'entraînement, les couches inférieures sont figées et ne changeront pas. Ensuite, toutes les couches sont débloquées et l'entraînement est fait sur l'entièreté du modèle. L'entraînement est fait sur une période de

60 époques pour la première partie de l’entraînement, suivi de 25 époques pour la deuxième partie. L’optimiseur *Adam* est utilisé avec un taux d’apprentissage de 0,001.

La comparaison du taux de reconnaissance des deux méthodes d’apprentissage permet de déterminer si certaines aptitudes à reconnaître des mouvements sont présentes dans le modèle entraîné sur le jeu de données d’activités et transférables sur le nouveau jeu de données.

3.5.4 RÉSULTATS ET DISCUSSION

La première étape consiste en l’analyse du taux de reconnaissance. Pour commencer, il est intéressant d’analyser les résultats enregistrés avec les radars UWB placés à 36 cm du sol. Puisqu’il s’agit d’un problème à quatre classes, un classeur aléatoire devrait avoir un taux de reconnaissance suivant approximativement la progression suivante : 0,25 (Top 1), 0,50 (Top 2) et 0,75 (Top 3). Un modèle qui est capable de classer les mouvements devrait avoir un taux de reconnaissance supérieur à un modèle aléatoire. Considérant le grand nombre de taux de reconnaissance qui ont été calculés, ces derniers sont consignés à l’Annexe B. Les différents taux de reconnaissance pour le modèle *Tuned EfficientNetB0* avec apprentissage à partir de zéro et par transfert, pour les capteurs installés à 36 cm du sol et 96 cm du sol, sont donnés dans les Figures B.1, B.2, B.3 et B.4. À des fins de synthèse, les taux moyens de reconnaissance sur l’ensemble des jeux de données LOSO sont inscrits dans le Tableau 3.7.

Afin de discuter de l’impact de la hauteur des radars UWB et de l’impact de l’apprentissage par transfert, il faut analyser les matrices de confusion des modèles. Avant de plonger

Modèle <i>Tuned EfficientNetB0</i>	Top 1 (%)	Top 2 (%)	Top 3 (%)
Apprentissage à partir de zéro (36 cm du sol)	56,1	79,8	91,0
Apprentissage à partir de zéro (96 cm du sol)	60,5	82,9	94,0
Apprentissage par transfert (36 cm du sol)	73,1	88,9	95,7
Apprentissage par transfert (96 cm du sol)	75,2	89,4	94,9

Tableau 3.7 : Comparaison du taux de reconnaissance moyen du modèle *Tuned EfficientNetB0* sur l'ensemble des jeux de données LOSO.

directement dans les matrices de confusion du modèle pour chaque LOSO, il est important d'établir à quoi pourrait ressembler une matrice de confusion d'un modèle qui est capable de faire correctement la distinction entre les quatre activités. Pour commencer, les quatre activités représentent une gradation d'une seule activité, soit *Bouger les Bras*. Ainsi, si les données sont révélatrices, il ne devrait pas y avoir de confusion entre la classe *Aucun mouvement* et *Mouvements grands*, car il s'agit de la plus grande disparité entre deux activités. Dans un même ordre d'idées, si les données sont suffisamment révélatrices et que le modèle est capable de bien distinguer entre l'activité *Aucun mouvement* et *Mouvements fins* (plus petite disparité entre deux activités), alors il devrait être capable de correctement classifier les autres activités, contenant beaucoup plus de mouvement. Ensuite, dans l'éventualité que le modèle fasse une mauvaise classification, il devrait confondre avec la classe la plus proche de cette dernière (*Aucun mouvement* avec *Mouvements fins*, *Mouvements intermédiaires* et *Mouvements grands*, etc.). Finalement, si les données sont révélatrices, le modèle devrait toujours tendre à faire les mêmes erreurs d'une instance à l'autre.

Les Figures C.1 à C.8, situées à l'Annexe C, contiennent les matrices de confusion du modèle *Tuned EfficientNetB0* entraîné à plusieurs reprises (5) sur les différents jeux de données LOSO pour les différentes combinaisons de hauteurs de capteurs et de méthodes d'apprentissage. Dans ces figures, les activités sont numérotées de 0 à 3 (*Aucun mouvement* à *Mouvements grands*). À des fins de synthèse, l'ensemble des matrices de confusion ont été colligées dans la Figure 3.24. La Figure contient le total des matrices de confusion pour chacune des combinaisons de méthodes d'apprentissage et de hauteurs de capteurs.

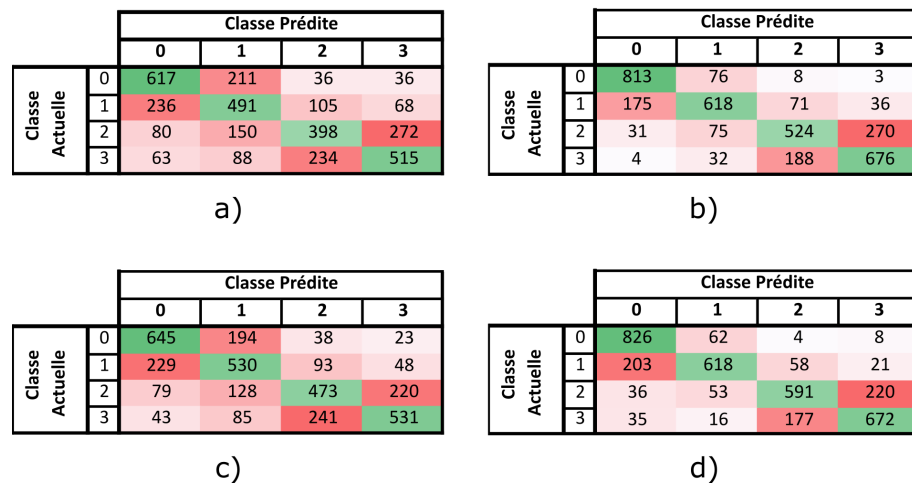


Figure 3.24 : Comparaisons des matrices de confusions de *Tuned EfficientNetB0* sommées pour l'entièreté des jeux de données LOSO, pour chacune des combinaisons de hauteurs de capteurs et de méthode d'apprentissage. a) Apprentissage à partir de zéro avec radars UWB placés à 36 cm du sol. b) Apprentissage par transfert avec radars UWB placés à 36 cm du sol. c) Apprentissage à partir de zéro avec radars UWB placés à 96 cm du sol. d) Apprentissage par transfert avec radars UWB placés à 96 cm du sol.

De prime abord, les résultats sont bons, mais pas optimaux. Les meilleurs résultats sont obtenus avec l'apprentissage par transfert avec un taux de reconnaissance de 75,2 % au Top 1. Puisque le taux de reconnaissance au Top 2 est 89,4 %, le modèle semble capable d'identifier

correctement le mouvement dans la plupart des cas, et dans le cas où il se trompe, hésite avec la bonne classe de mouvement. Considérant qu'une grande différence visuelle existe entre les différents mouvements, on pourrait s'attendre à un meilleur taux de reconnaissance. Comme discuté précédemment, la méthode de prétraitement n'est peut-être pas idéale et accentue trop les petites variations, rendant plus difficile la reconnaissance de la gradation des mouvements. En se basant sur la Figure 3.24 et sur les caractéristiques des matrices de confusion énumérées plus tôt, il est possible de constater que globalement, les modèles sont bel et bien capables d'extraire les caractéristiques clés des mouvements. Premièrement, les mouvements qui sont le moins confondus sont *Aucun mouvement* et *Mouvements grands*. Ensuite, il est possible de constater que la plupart des confusions surviennent entre *Aucun mouvement* et *Mouvements fins*, et *Mouvements intermédiaires* et *Mouvements grands*.

En ce qui concerne l'impact de la méthode d'apprentissage, il est possible de constater que pour les deux hauteurs, le taux de reconnaissance est significativement meilleur lorsque l'apprentissage par transfert est utilisé. Il est donc possible de conclure que le modèle, lorsqu'il est entraîné sur le jeu de données d'activités, est capable d'extraire les caractéristiques relatives au mouvement des bras, en plus du mouvement du corps entier. Les limitations du modèle à reconnaître certaines activités ne peuvent donc pas être blâmées uniquement sur l'incapacité du modèle à voir les mouvements des membres du corps. Il est important de souligner qu'il se peut tout de même qu'une partie de ces limitations provienne du fait que le modèle n'est pas capable de classer systématiquement le type de mouvement.

En ce qui concerne l'impact de la hauteur des capteurs sur la capacité du modèle à

classifier les mouvements de bras, placer les radars UWB plus près du centre de gravité d'une personne (soit environ 1m) affecte favorablement le taux de reconnaissance. Une augmentation de 4,4 % du taux de reconnaissance au Top 1 est présent pour l'apprentissage à partir de zéro et 2,1 % pour l'apprentissage par transfert. Dans le but d'améliorer la reconnaissance dans des travaux subséquents, il serait intéressant de placer les radars UWB à environ 1 m.

Un des objectifs de cette expérimentation était d'identifier les endroits dans l'appartement où les mouvements sont le mieux reconnus. La Figure 3.25 contient les taux de reconnaissance du modèle *Tuned EfficientNetB0* pour chacune des combinaisons de hauteurs de capteurs et de méthode d'apprentissage en fonction de la position où se déroulent les mouvements de bras. Il est intéressant de constater que les mouvements effectués aux positions 1, 2, 4 et 8 sont généralement les moins bien reconnus. Ces positions sont près des capteurs et souvent dans le champ de vision d'un seul capteur. À l'inverse, les mouvements effectués aux positions 3, 6, 7, et 9 sont généralement mieux reconnus. Ces dernières positions sont plus loin des radars UWB et dans le champ de vision de plusieurs radars UWB. La thèse développée plus tôt disant que la faible performance du modèle *Tuned EfficientNetB0* sur les activités *Boire* et *Prendre des médicaments* pourrait être reliée à certains angles morts dans l'appartement, notamment la salle de bains, perd un peu de crédibilité lorsque l'on analyse les matrices de confusion du modèle sur les jeux de données LOSO 10 et 12 à la Figure C.6, soit devant l'évier de la salle de bains. Même si le taux de reconnaissance à ces positions est loin d'être parfait, les matrices de confusion indiquent que le modèle peut faire la distinction entre *Aucun mouvement* et *Mouvements fins*. Dans le but d'améliorer la reconnaissance dans des

travaux subséquents, il serait intéressant d’explorer l’utilisation d’un plus grand nombre de radars UWB pour la reconnaissance d’activités.

H1 - Apprentissage à partir de zéro		H2 - Apprentissage à partir de zéro		H1 - Apprentissage par transfert		H2 - Apprentissage par transfert	
Position	Taux de Reconnaissance	Position	Taux de Reconnaissance	Position	Taux de Reconnaissance	Position	Taux de Reconnaissance
5	0.67	9 - Fermé	0.69	7	0.90	9 - Fermé	0.92
6	0.65	10 - Ouvert	0.67	9 - Ouvert	0.86	7	0.88
7	0.62	9 - Ouvert	0.67	6	0.84	9 - Ouvert	0.87
3	0.60	3	0.66	10 - Fermé	0.83	1	0.80
9 - Fermé	0.58	6	0.65	9 - Fermé	0.79	2	0.79
9 - Ouvert	0.58	5	0.62	3	0.76	6	0.76
10 - Fermé	0.58	7	0.60	5	0.70	3	0.74
10 - Ouvert	0.52	2	0.59	1	0.68	5	0.73
1	0.51	1	0.59	10 - Ouvert	0.66	4	0.72
4	0.50	4	0.57	2	0.64	10 - Ouvert	0.70
2	0.47	10 - Fermé	0.51	4	0.64	10 - Fermé	0.67
8	0.45	8	0.44	8	0.46	8	0.45

Figure 3.25 : Comparaisons du taux de reconnaissance de *Tuned EfficientNetB0* pour chacune des combinaisons de hauteurs de capteurs et de méthode d’apprentissage en fonction de la position où se déroule les mouvements de bras. Les positions 9 et 10 ouvert correspondent aux LOSO 9 et 10, et les positions 9 et 10 fermé correspondent aux LOSO 11 et 12.

Un dernier point touché par l’expérimentation était de voir l’impact d’une porte comme obstacle sur la reconnaissance de mouvements. La position 9 montre bien que peu importe si la porte est ouverte ou fermée, le taux de reconnaissance à cette position reste semblable.

Le but de cette expérience était d’établir d’où provenaient les problèmes de reconnaissances d’activités. À cause de la nature opaque des réseaux de neurones, il est impossible d’identifier avec certitude la source des limitations, mais certaines théories peuvent être avancées. En ce qui concerne la capacité des radars UWB et de la méthode de prétraitement des données à extraire les mouvements clés comme le mouvement des mains et des bras, il est possible de dire qu’ils sont adéquats et permettent de mettre en évidence ces mouvements. Ceci est soutenu par la visualisation à l’œil nu de différences au niveau des données prétraitées et

des performances adéquates du modèle développé sur le jeu de données. Avec l'analyse du taux de reconnaissance des mouvements aux différentes positions, il a aussi été possible d'identifier les endroits où la reconnaissance de mouvements est favorisée. Ces zones correspondent généralement aux endroits où les champs de vision des radars UWB se superposent. L'inverse est aussi vrai. Les endroits associés à de moins bons taux de reconnaissance sont couverts par le champ de vision d'un seul radar UWB ou près de la limite de leurs champs de vision. Il s'agit des endroits près des murs où se trouvent les radars UWB. La hauteur des radars UWB s'est avérée un facteur important pour la reconnaissance d'activités. Un radar UWB placé près du centre de gravité d'une personne permet d'améliorer le taux de reconnaissance.

Cette expérimentation laisse place à de nouveaux questionnements. Par exemple, il a été montré que la méthode de prétraitement de données combinée au modèle développé permet de reconnaître les mouvements des bras. Cependant, les mouvements sont simples et répétés en continu pour la durée de l'enregistrement. Une AVQ contient ces mouvements, mais ceux-ci sont entremêlés et parsemés. Il n'est donc pas possible avec cette expérimentation de vérifier que des mouvements de courte durée peuvent être reconnus à travers d'autres mouvements. Dans un même ordre d'idées, il est impossible de valider que les lacunes de reconnaissance d'activités de *Tuned EfficientNetB0* ne proviennent pas d'une incapacité de donner du sens à la chronologie des mouvements, ou simplement du manque de données dans le jeu de données.

CHAPITRE IV

CONCLUSION

Un des enjeux sociaux à surveiller dans les prochaines années est le vieillissement de la population. Ce qui est problématique avec ce vieillissement est la quantité de ressources requises pour assister cette population vulnérable dans leur quotidien. Cette problématique entraîne le besoin pour de nouvelles solutions intelligentes pour assister cette population avec plus de facilité. Les habitats intelligents ont généré beaucoup d'intérêt pour répondre à ce problème, puisqu'ils peuvent être utilisés pour détecter des événements graves et suivre le bon déroulement des AVQ. L'objectif de ce mémoire était de déterminer comment faire la reconnaissance d'activités dans un environnement intelligent avec des radars UWB. Pour ce faire, deux approches ont été explorées, soit la combinaison de réseaux de neurones et de systèmes experts, et le développement d'un réseau de neurones CNN pour la reconnaissance d'AVQ. De plus, une expérimentation supplémentaire a été réalisée dans le but de caractériser les faiblesses des solutions développées.

4.1 REVUE DES CONTRIBUTIONS

Dans un premier temps, la combinaison de réseaux de neurones avec un système expert dans le but d'améliorer le taux de reconnaissance du réseau de neurones a été explorée. Pour ce faire, un modèle central développé par Maitre *et al.* (2021), a été combiné à un système expert. Les limitations fondamentales de cette approche ont été rapidement identifiées : difficulté

d'extraire des connaissances expertes des données brutes, angles morts du modèle central d'un jeu de données à l'autre, haute variabilité d'une activité à l'autre, rigidité des règles. Le taux de reconnaissance offert par la solution globale est parfois légèrement meilleur au Top 1, mais est plus souvent moins bon dans le reste des Top-N. Cette approche a permis de souligner l'importance d'un modèle central qui offre un meilleur taux de reconnaissance par lui-même. Il ne s'agit pas ici d'une contribution majeure, mais d'une approche qui a été explorée dans le cadre du mémoire.

Dans un deuxième temps, un réseau de neurones basé sur EfficientNetB0 (Tan et Le, 2019), appelé *Tuned EfficientNetB0 avec LSTM*, a été développé. Quelques améliorations ont été apportées à l'architecture originale en optimisant la taille et la forme des filtres dans les couches CNN, en plus de fusionner les informations sur la position calculée de la personne dans l'environnement avec des couches LSTM. *Tuned EfficientNet avec LSTM* offre un meilleur taux de reconnaissance que le réseau de neurones développé par Maitre *et al.* (2021), utilisant le même jeu de données (augmentation de 18,63 % sur le taux de reconnaissance moyen au Top 1 et 29,08 % sur le taux de reconnaissance balancé du Top-1). Il s'agit ici de la contribution principale proposée par ce mémoire.

Pour terminer, un nouveau jeu de données a été construit et enregistré au LIARA dans le but d'identifier la source des difficultés du modèle à reconnaître certaines activités, soit les activités contenant moins de déplacements et pouvant se dérouler dans la salle de bain. Le nouveau jeu de données contient 4 activités de mouvements de bras (*Aucun Mouvement*, *Mouvements Fins*, *Mouvements Intermédiaires* et *Mouvements Grands*) effectuées à 10 diffé-

rentes positions dans l'environnement. L'analyse des données brutes et prétraitées, en plus du taux de reconnaissance du modèle *Tuned EfficientNetB0* entraîné sur ce nouveau jeu de données, a permis d'identifier les endroits dans l'appartement intelligent du LIARA où la reconnaissance est plus difficile et de mesurer l'impact de la hauteur des radars UWB dans l'environnement. Les positions couvertes par le champ de vision d'un seul radar UWB sont les endroits les moins bien captés dans l'environnement, c'est-à-dire près des murs où les radars UWB sont installés. Il a aussi été montré que la hauteur des radars UWB a un impact non négligeable sur la capacité du modèle à percevoir les mouvements. Les meilleurs résultats ont été atteints avec les capteurs installés à 96 cm du sol. Finalement, il a été montré que le modèle *Tuned EfficientNetB0* entraîné sur le jeu de données d'activités est capable de plus facilement reconnaître les mouvements que le même modèle entraîné à partir de zéro.

4.2 LIMITES

Dans le cadre de ce mémoire, certaines limitations ont été identifiées.

En ce qui concerne l'utilisation de systèmes experts, ils ne semblent pas être une approche adéquate pour améliorer le taux de reconnaissance d'un réseau de neurones avec seulement les données de radars UWB. De meilleurs gains peuvent être atteints avec un meilleur classifieur.

Ensuite, le jeu de données utilisé initialement contient certaines limitations fondamentales qui doivent être adressées dans le futur afin de développer de meilleures solutions. Les

limitations principales proviennent entre autres du nombre restreint d'instances. Considérant le nombre de façons possibles d'exécuter une activité, comme l'activité *Cuisiner*, le nombre d'instances d'activités présentes dans le jeu de données est assez restreint. Un plus grand nombre d'instances d'activités pourrait aider les modèles à mieux généraliser. Dans un même ordre d'idées, puisque l'ensemble des mesures ont été faites dans un seul environnement, l'intégration d'autres environnements dans le jeu de données pourrait encourager un modèle à extraire le caractère général d'une activité, et non ce qu'est cette activité au LIARA. Finalement, un plus large spectre d'activités pourrait permettre de développer des solutions plus complètes.

Une autre limitation provient de la façon dont les radars UWB sont utilisés. Les expérimentations ont montré que le taux de reconnaissance est meilleur lorsque les radars UWB sont placés à 96 cm du sol. Les expérimentations futures devraient donc prioriser le placement à une hauteur de 1 m. De plus, les endroits dans l'appartement intelligent du LIARA où la reconnaissance est plus difficile correspondent aux endroits couverts par le champ de vision d'un seul radar UWB. Il serait donc intéressant d'utiliser un plus grand nombre de radars UWB dans l'environnement.

4.3 TRAVAUX FUTURS

Plusieurs travaux futurs peuvent découler de ce travail. Dans un premier temps, il est possible d'adresser directement les limitations énumérées plus haut. Ainsi, l'augmentation du jeu de données, en termes de nombre d'instances, d'environnements et de type d'activités, est

à considérer. Une autre piste d'amélioration pour les travaux futurs consiste à mieux placer les radars UWB et d'augmenter le nombre de radars UWB.

En ce qui concerne le modèle développé, une architecture déjà existante a été exploitée. Dans un premier temps, il serait intéressant d'explorer l'utilisation d'autres architectures et d'autres techniques d'apprentissage plus avancées. Par exemple, le mécanisme *Attention* (Vaswani *et al.*, 2017) est une alternative nouvelle aux couches LSTM qui ont généré beaucoup d'intérêt dernièrement. Considérant la taille présentement réduite du jeu de données d'activités, et en prévision du besoin de réentraînement de la solution lors du déploiement dans un nouvel environnement, des techniques de *Few Shot Learning* sont à considérer. Les possibilités sont vastes, mais une approche intéressante serait de décomposer le problème de la reconnaissance d'activités en plusieurs tâches de reconnaissance et d'effectuer du *Multi-Task Learning* (Caruana, 1997) dans le but que la reconnaissance d'une de ces sous-activités aide à la reconnaissance d'une autre. Dans ce scénario, il serait possible de séparer la reconnaissance d'activités en identification de la salle où se trouve la personne, identification de certains mouvements clés (comme *Déplacement*, *Aucun Mouvement* et *Grands Mouvements*) et de l'identification de l'activité. Dans ce scénario, même si la reconnaissance d'activités n'est pas meilleure, il serait possible d'exploiter les autres sorties du modèle.

BIBLIOGRAPHIE

Aggarwal, C. C. (2018). *Neural Networks and Deep Learning : A Textbook*. Cham, Suisse : Springer International Publishing. http://dx.doi.org/10.1007/978-3-319-94463-0_3

Alam, M. R., Reaz, M. B. I. et Mohd Ali, M. (2012). A review of smart homes – past, present, and future. *IEEE Transactions on Systems, Man, and Cybernetics Part C : Applications and Reviews*, 42, 1190–1203. <http://dx.doi.org/10.1109/TSMCC.2012.2189204>

Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, S., Al-Ammar, M. et Al-Khalifa, H. (2016). Ultra wideband indoor positioning technologies : Analysis and recent advances. *Sensors*, 16(5), 707. <http://dx.doi.org/10.3390/s16050707>

Alshammari, S. A., Alhassan, Abdullatif M. and Aldawsari, M. A., Bazuhair, F. O., Alotaibi, F. K., Aldakhil, A. A. et Abdulfattah, F. W. (2018). Falls among elderly and its relation with their health problems and surrounding environmental factors in riadh. *Journal of family & community medicine*, 25(1), 29–34. http://dx.doi.org/10.4103/jfcm.JFCM_48_17

Alzheimer Society of Canada (2016). Report summary. prevalence and monetary costs of dementia in canada (2016) : A report by the alzheimer society of canada. *Health promotion and chronic disease prevention in Canada : Research, policy and practice*, 36(10), 231–232. <http://dx.doi.org/10.24095/hpcdp.36.10.04>

Alzheimer's Association (2018). 2018 alzheimer's disease facts and figures. *Alzheimer's & Dementia*, 14(3), 367–429. <http://dx.doi.org/10.1016/j.jalz.2018.02.001>

Ben-David, S. et Schuller, R. (2003). Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines* 567–580. Heidelberg, Allemagne : Springer

Bochkovskiy, A., Wang, C.-Y. et Liao, H.-y. (2020). Yolov4 : Optimal speed and accuracy of object detection. Récupéré le 2021-05-10 de <https://arxiv.org/abs/2004.10934>

Bouchard, B., Bouchard, K. et Bouzouane, A. (2020a). A smart cooking device for assisting cognitively impaired users. *Journal of Reliable Intelligent Environments*, 6(2), 107–125. <http://dx.doi.org/10.1007/s40860-020-00104-3>

Bouchard, K., Maitre, J., Bertuglia, C. et Gaboury, S. (2020b). Activity recognition in smart homes using uwb radars. *Procedia Computer Science*, 170, 10–17. <http://dx.doi.org/10.1016/j.procs.2020.03.004>

Broomhead, D. et King, G. (1986). On the qualitative analysis of experimental dynamical systems. *Physica D : Nonlinear Phenomena*, 20, 217–236. [http://dx.doi.org/10.1016/0167-2789\(86\)90031-X](http://dx.doi.org/10.1016/0167-2789(86)90031-X)

Brunner, G., Melnyk, D., Sigfússon, B. et Wattenhofer, R. (2019). Swimming style recognition and lap counting using a smartwatch and deep learning. Dans *Proceedings of the 23rd International Symposium on Wearable Computers*, ISWC '19, p. 23–31., New York, NY, USA. Association for Computing Machinery. <http://dx.doi.org/10.1145/3341163.3347719>

Caruana, R. (1997). Multitask learning. *Machine Learning*, 28(1), 41–75. <http://dx.doi.org/10.1023/A:1007379606734>

Chollet, F. (2016). Xception : Deep learning with depthwise separable convolutions. *CoRR*, *abs/1610.02357*. Récupéré le 2021-05-10 de <http://arxiv.org/abs/1610.02357>

Cook, D. J. et Das, S. K. (2004). *Overview*, Dans *Smart Environments*, (p. 1–10). John Wiley & Sons, Ltd

Cortes, C. et Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273–297. <http://dx.doi.org/10.1007/BF00994018>

Cowell, R. G., Dawid, A. P., Lauritzen, S. L. et Spiegelhalter, D. J. (1998). *Logic, Uncertainty, and Probability*, Dans *Probabilistic Networks and Expert Systems*, (p. 5–23). Springer : New York, NY

Cristofani, E., Friederich, F., Vandewal, M. et Jonuscheit, J. (2016). *Advanced Ultrawide-band Radar*. CRC Press

Ding, C., Zhang, L., Gu, C., Bai, L., Liao, Z., Hong, H., Li, Y. et Zhu, X. (2018). Non-contact human motion recognition based on uwb radar. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 8(2), 306–315. <http://dx.doi.org/10.1109/>

Doshi-Velez, F. et Kim, B. (2017). Towards a rigorous science of interpretable machine learning. Récupéré le 2021-05-10 de <https://arxiv.org/abs/1702.08608>

Du, H., Jin, T., He, Y., Song, Y. et Dai, Y. (2019). Segmented convolutional gated recurrent neural networks for human activity recognition in ultra-wideband radar. *Neurocomputing*. <http://dx.doi.org/10.1016/j.neucom.2018.11.109>

Ercolano, G., Riccio, D. et Rossi, S. (2017). Two deep approaches for adl recognition : A multi-scale lstm and a cnn-lstm with a 3d matrix skeleton representation. Dans *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 877–882. <http://dx.doi.org/10.1109/ROMAN.2017.8172406>

Espinilla, M., Medina, J. et Nugent, C. (2018). UCAMl cup. analyzing the UJA human activity recognition dataset of activities of daily living. *Proceedings*, 2(19), 1267. <http://dx.doi.org/10.3390/proceedings2191267>

Fillit, H. M., Butler, R. N., O’Connell, A. W., Albert, M. S., Birren, J. E., Cotman, C. W., Greenough, W. T., Gold, P. E., Kramer, A. F., Kuller, L. H., Perls, T. T., Sahagan, B. G. et Tully, T. (2002). Achieving and maintaining cognitive vitality with aging. *Mayo Clinic Proceedings*, 77(7), 681–696. <http://dx.doi.org/10.4065/77.7.681>

Fortin-Simard, D., Bilodeau, J., Bouchard, K., Gaboury, S., Bouchard, B. et Bouzouane, A. (2015). Exploiting passive rfid technology for activity recognition in smart homes. *IEEE Intelligent Systems*, 30(4), 7–15. <http://dx.doi.org/10.1109/MIS.2015.18>

Francillette, Y., Bouchard, B., Boucher, E., Gaboury, S., Bernard, P., Romain, A. J. et Bouchard, K. (2018). Development of an exergame on mobile phones to increase physical activity for adults with severe mental illness. Dans *Proceedings of the 11th Pervasive Technologies Related to Assistive Environments Conference, PETRA '18*, p. 241–248., New York, NY, USA. Association for Computing Machinery. <http://dx.doi.org/10.1145/3197768.3201521>

Fukushima, K. (1980). Neocognitron : A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4), 193–202. <http://dx.doi.org/10.1007/BF00344251>

Garcez, A., Broda, K. et Gabbay, D. (2001). Symbolic knowledge extraction from trained neural networks : A sound approach. *Artificial Intelligence*, 125, 155–207. [http://dx.doi.org/10.1016/S0004-3702\(00\)00077-1](http://dx.doi.org/10.1016/S0004-3702(00)00077-1)

Giarratano, J. (2005). *Expert systems : principles and programming*. Boston, MA, USA : Thomson Course Technology.

Giroux, S., Bauchet, J., Pigot, H., Lussier-Desrochers, D. et Lachappelle, Y. (2008). Pervasive behavior tracking for cognitive assistance. Dans *Proceedings of the 1st International Conference on Pervasive Technologies Related to Assistive Environments*, PETRA '08, New York, NY, USA. Association for Computing Machinery. <http://dx.doi.org/10.1145/1389586.1389684>

Glorot, X., Bordes, A. et Bengio, Y. (2011). Deep sparse rectifier neural networks. Dans G. Gordon, D. Dunson, et M. Dudík (dir.). *Proceedings of Machine Learning Research*, volume 15, 315–323., Fort Lauderdale, FL, USA. Journal of Machine Learning Research Workshop and Conference Proceedings. Récupéré le 2021-05-10 de <http://proceedings.mlr.press/v15/glorot11a.html>

Goodfellow, I., Bengio, Y. et Courville, A. (2016). *Deep Learning*. Cambridge, MA, USA : MIT Press. <http://www.deeplearningbook.org>.

Graves, A., Jaitly, N. et rahman Mohamed, A. (2013). Hybrid speech recognition with deep bidirectional LSTM. Dans *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE. <http://dx.doi.org/10.1109/asru.2013.6707742>

Hamilton, I. A. (2020). An AI tool which reconstructed a pixelated picture of barack obama to look like a white man perfectly illustrates racial bias in algorithms. Récupéré le 2021-05-10 de <https://www.businessinsider.com/depixelator-turned-obama-white-illustrates-racial-bias-in-ai-2020-6>

Harrisburg University of Science and Technology (2020). Hu facial recognition software predicts criminality. Récupéré le 2021-05-10 de <http://archive.is/N1HVe>

Hastie, T., Tibshirani, R. et Friedman, J. (2009). *The elements of statistical learning : data mining, inference and prediction* (2 éd.). Verlag, NY, USA : Springer. <http://dx.doi.org/10.1007/978-0-387-84858-7>

Hatzilygeroudis, I. et Prentzas, J. (2004). Integrating (rules, neural networks) and cases for knowledge representation and reasoning in expert systems. *Expert Systems with Applications*, 27(1), 63–75. <http://dx.doi.org/10.1016/j.eswa.2003.12.004>

He, K., Zhang, X., Ren, S. et Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, *abs/1512.03385*. Récupéré le 2021-05-10 de <http://arxiv.org/abs/1512.03385>

Heideman, M., Johnson, D. et Burrus, C. (1985). Gauss and the history of the fast fourier transform. *Archive for History of Exact Sciences*, 34, 265–277. <http://dx.doi.org/10.1007/BF00348431>

Ho, T. K. (1995). Random decision forests. Dans *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1)-Volume 1*, ICDAR '95, p. 278., Montreal, QC, Canada. IEEE Computer Society. <http://dx.doi.org/10.1109/ICDAR.1995.598994>

Hochreiter, S. et Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9, 1735–80. <http://dx.doi.org/10.1162/neco.1997.9.8.1735>

Hubel, D. H. et Wiesel, T. N. (1959). Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, 148(3), 574–591. <http://dx.doi.org/10.1113/jphysiol.1959.sp006308>

Irvine, N., Nugent, C., Zhang, S., Wang, H. et NG, W. W. Y. (2019). Neural network ensembles for sensor-based human activity recognition within smart environments. *Sensors*, 20(1), 216. <http://dx.doi.org/10.3390/s20010216>

Jenssen, R. O., Eckerstorfer, M., Jacobsen, S. et Storvold, R. (2018). Drone-mounted uwb radar system for measuring snowpack properties : Technical implementation, specifications and initial results. Dans *International Snow Science Workshop*.

Jokanovic, B., Amin, M. et Erol, B. (2017). Multiple joint-variable domains recognition of human motion. Dans *2017 IEEE Radar Conference (RadarConf)*, 0948–0952. <http://dx.doi.org/10.1109/RADAR.2017.7944340>

Kenny, G. P., Yardley, J. E., Martineau, L. et Jay, O. (2008). Physical work capacity in older adults : Implications for the aging worker. *American Journal of Industrial Medicine*, 51(8), 610–625. <http://dx.doi.org/10.1002/ajim.20600>

Kingma, D. P. et Ba, J. (2017). Adam : A method for stochastic optimization. Récupéré le 2021-05-10 de <https://arxiv.org/abs/1412.6980>

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Mallocci, M., Kolesnikov, A., Duerig, T. et Ferrari, V. (2020). The open images dataset v4 : Unified image classification, object detection, and visual relationship detection at scale. *International Journal of Computer Vision*. <http://dx.doi.org/10.1007/s11263-020-01316-z>

Lang, Y., Hou, C., Yang, Y., Huang, D. et He, Y. (2017). Convolutional neural network for human micro-doppler classification. European Microwave Conference.

Le, M., Le, D.-K. et Lee, J. (2020). Multivariate singular spectral analysis for heartbeat extraction in remote sensing of uwb impulse radar. *Sensors and Actuators A : Physical*, 306. <http://dx.doi.org/10.1016/j.sna.2020.111968>

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W. et Jackel, L. (1990). Handwritten digit recognition with a back-propagation network. Dans D. Touretzky (dir.). *Advances in Neural Information Processing Systems*, volume 2, 396–404. Morgan-Kaufmann. Récupéré le 2021-05-10 de <http://yann.lecun.com/exdb/publis/pdf/lecun-90c.pdf>

Lee, I. et Lee, K. (2015). The internet of things (iot) : Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4), 431–440. <http://dx.doi.org/10.1016/j.bushor.2015.03.008>

Leonardis, G. D., Rosati, S., Balestra, G., Agostini, V., Panero, E., Gastaldi, L. et Knaflitz, M. (2018). Human activity recognition by wearable sensors : Comparison of different classifiers for real-time applications. Dans *2018 IEEE International Symposium on Medical Measurements and Applications (MeMeA)*. IEEE. <http://dx.doi.org/10.1109/memea.2018.8438750>

Li, H.-J. et Kiang, Y.-W. (2005). 10-radar and inverse scattering. In W.-K. CHEN (dir.),

The Electrical Engineering Handbook 671–690. Burlington : Academic Press

Liang, S. D. (2016). Sense-through-wall human detection based on uwb radar sensors. *Signal Processing*, 126, 117–124. <http://dx.doi.org/10.1016/j.sigpro.2015.09.022>

Ling, S. (2017). *University physics*, (p. 882–889). XanEdu OpenStax, Rice University : Acton, MA Houston, Texas.

Ma, L., Liu, M., Wang, N., Wang, L., Yang, Y. et Wang, H. (2020). Room-level fall detection based on ultra-wideband (uwb) monostatic radar and convolutional long short-term memory (Istm). *Sensors*, 20(4), 1105. <http://dx.doi.org/10.3390/s20041105>

Madhuranga, D., Madushan, R., Siriwardane, C. et Gunasekera, K. (2020). Real-time multimodal ADL recognition using convolution neural networks. *The Visual Computer*. <http://dx.doi.org/10.1007/s00371-020-01864-y>

Maitre, J., Bouchard, K., Bertuglia, C. et Gaboury, S. (2021). Recognizing activities of daily living from uwb radars and deep learning. *Expert Systems with Applications*, 164, 113994. <http://dx.doi.org/10.1016/j.eswa.2020.113994>

Maitre, J., Bouchard, K. et Gaboury, S. (2020). Fall detection with UWB radars and CNN-LSTM architecture. *IEEE Journal of Biomedical and Health Informatics*. <http://dx.doi.org/10.1109/jbhi.2020.3027967>

Malajner, M., Šipoš, D. et Gleich, D. (2020). Design of a low-cost ultra-wide-band radar platform. *Sensors (Basel, Switzerland)*, 20(10), 2867. <http://dx.doi.org/10.3390/s20102867>

McCulloch, Warren S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115–133. <http://dx.doi.org/10.1007/BF02478259>

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. Cambridge, MA, USA : MIT Press.

Mitchell, T. (1997). *Machine Learning*. New York, NY, USA : McGraw-Hill.

Nisar, M. A., Shirahama, K., Li, F., Huang, X. et Grzegorzec, M. (2020). Rank pooling approach for wearable sensor-based adls recognition. *Sensors*, 20(12), 3463. <http://dx.doi.org/10.3390/s20123463>

Noelker, L. et Browdie, R. (2013). Sidney katz, md : A new paradigm for chronic illness and long-term care. *The Gerontologist*, 54. <http://dx.doi.org/10.1093/geront/gnt086>

Novelda AS (2018). *X4M200 Datasheet*. Novelda AS. Rev. E

Offermann-van Heek, J., Schomakers, E.-M. et Ziefle, M. (2019). Bare necessities? how the need for care modulates the acceptance of ambient assisted living technologies. *International Journal of Medical Informatics*, 127, 147–156. <http://dx.doi.org/10.1016/j.ijmedinf.2019.04.025>

Pérez, R. A. M. (2017). Experta. <https://github.com/nilp0inter/experta>.

Prince, M., Wimo, A., Guerchet, M., Ali, G.-C., Wu, Y.-T. et Prina, M. (2015). World alzheimer report 2015. the global impact of dementia : An analysis of prevalence, incidence, cost and trends. Récupéré le 2021-05-10 de <https://www.alzint.org/resource/world-alzheimer-report-2015/>

Ramachandran, P., Zoph, B. et Le, Q. V. (2017a). Searching for activation functions. *CoRR*, abs/1710.05941. Récupéré le 2021-05-10 de <http://arxiv.org/abs/1710.05941>

Ramachandran, P., Zoph, B. et Le, Q. V. (2017b). Swish : a self-gated activation function. *arXiv : Neural and Evolutionary Computing*. Récupéré le 2021-05-10 de <https://arxiv.org/abs/1710.05941>

Remi-Omosowon, A. et Pyeasgya, Y. G. (2014). A simple and easy-to-use implementation of a genetic algorithm library in python. <https://github.com/remiomosowon/pyeasgya>.

Ren, L., Koo, Y. S., Wang, Y. et Fathy, A. E. (2015). Noncontact heartbeat detection using

uwb impulse doppler radar. Dans *2015 IEEE Topical Conference on Biomedical Wireless Technologies, Networks, and Sensing Systems (BioWireless)*, 1–3. <http://dx.doi.org/10.1109/BIOWIRELESS.2015.7152128>

Ren, L., Wang, H., Naishadham, K., Kilic, O. et Fathy, A. E. (2016). Phase-based methods for heart rate detection using uwb impulse doppler radar. *IEEE Transactions on Microwave Theory and Techniques*, 64(10), 3319–3331. <http://dx.doi.org/10.1109/TMTT.2016.2597824>

Rokach, L. (2009). Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2), 1–39. <http://dx.doi.org/10.1007/s10462-009-9124-7>

Rosenblatt, F. (1958). The perceptron : A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65–386.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. et Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 211–252. <http://dx.doi.org/10.1007/s11263-015-0816-y>

Russell, S. J. et Norvig, P. (2010). *Artificial intelligence : a modern approach* (3 éd.). Upper Saddle River, NJ, USA : Prentice Hall.

Sadrezami, H., Bolic, M. et Rajan, S. (2020). Fall detection using standoff radar-based sensing and deep convolutional neural network. *IEEE Transactions on Circuits and Systems II : Express Briefs*, 67(1), 197–201. <http://dx.doi.org/10.1109/TCSII.2019.2904498>

Simonyan, K. et Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. Récupéré le 2021-05-10 de <http://arxiv.org/abs/1409.1556>

Singh, S., Liang, Q., Chen, D. et Sheng, L. (2011). Sense through wall human detection using uwb radar. *Eurasip Journal on Wireless Communications and Networking-EURASIP J WIREL COMMUN NETW*, 2011, 1–11. <http://dx.doi.org/10.1186/1687-1499-2011-20>

Steger, H., Wörtge, M., Siegmund, G. et Rembe, C. (2006). Measuring MEMS in motion by laser doppler vibrometry. In *Optical Science and Engineering* 245–292. CRC Press

Sukor, A. S. A., Zakaria, A., Rahim, N. A. et Setchi, R. (2018). Semantic knowledge base in support of activity recognition in smart home environments. *International Journal of Engineering & Technology*, 7(4.27), 67–72. <http://dx.doi.org/10.14419/ijet.v7i4.27.22482>

Sukthankar, G., Geib, C., Bui, H. H., Pynadath, D. et Goldman, R. P. (2014). An introduction to plan, activity, and intent recognition. In *Plan, Activity, and Intent Recognition : Theory and Practice*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., (1st éd.).

Tan, M. et Le, Q. V. (2019). Efficientnet : Rethinking model scaling for convolutional neural networks. Dans *Proceedings of the 36th International Conference on Machine Learning*, volume 97 de *Proceedings of Machine Learning Research*. Récupéré le 2021-05-10 de <http://proceedings.mlr.press/v97/tan19a/tan19a.pdf>

Taylor, J. (2012). *Ultrawideband Radar*. Boca Raton, FL : Taylor & Francis.

Taylor, J. D. (2000). *Ultra-wideband radar technology*. Boca Raton, FL, USA : CRC Press. <http://dx.doi.org/10.1201/9781003068112>

The Sentencing Project (2018). Regarding racial disparities in the united states criminal justice system. Récupéré le 2021-05-10 de <https://www.sentencingproject.org/publications/un-report-on-racial-disparities/>

Tiemann, J. et Wietfeld, C. (2017). Scalable and precise multi-uav indoor navigation using tdoa-based uwb localization. Dans *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 1–7. <http://dx.doi.org/10.1109/IPIN.2017.8115937>

United Nations Department of Economic and Social Affairs (2019). World population prospect 2019 : Highlights. Récupéré le 2021-05-10 de https://population.un.org/wpp/Publications/Files/WPP2019_Highlights.pdf

Valenzuela, L. M. A., Bentley, J. M. et Lorenz, R. D. (2004). Expert system for integrated

control and supervision of dry-end sections of paper machines. *IEEE Transactions on Industry Applications*, 40(2), 680–691. <http://dx.doi.org/10.1109/TIA.2004.824507>

van Weel, C., Vermeulen, H. et van den Bosch, W. (1995). Falls, a community care perspective. *The Lancet*, 345(8964), 1549–1551. [http://dx.doi.org/10.1016/S0140-6736\(95\)91091-3](http://dx.doi.org/10.1016/S0140-6736(95)91091-3)

Vasara, P. (2015). Not ageing in place : Negotiating meanings of residency in age-related housing. *Journal of Aging Studies*, 35, 55–64. <http://dx.doi.org/10.1016/j.jaging.2015.07.004>

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. et Polosukhin, I. (2017). Attention is all you need. *CoRR*, abs/1706.03762. Récupéré le 2021-05-10 de <http://arxiv.org/abs/1706.03762>

Voulodimos, A., Doulamis, N., Doulamis, A. et Protopapadakis, E. (2018). Deep learning for computer vision : A brief review. *Computational Intelligence and Neuroscience*, 2018. <http://dx.doi.org/10.1155/2018/7068349>

Wang, Y. et Yao, Q. (2019). Few-shot learning : A survey. *CoRR*, abs/1904.05046. Récupéré le 2021-05-10 de <http://arxiv.org/abs/1904.05046>

Werbos, P. (1974). *Beyond Regression : New Tools for Prediction and Analysis in the Behavioral Science*. (Thèse de doctorat). Harvard University.

Yan, J., Hong, H., Zhao, H., Li, Y., Gu, C. et Zhu, X. (2016). Through-wall multiple targets vital signs tracking based on vmd algorithm. *Sensors*, 16. <http://dx.doi.org/10.3390/s16081293>

Yang, Y., Hou, C., Lang, Y., Guan, D., Huang, D. et Xu, J. (2019). Open-set human activity recognition based on micro-doppler signatures. *Pattern Recognition*, 85, 60–69. <http://dx.doi.org/10.1016/j.patcog.2018.07.030>

Zafari, F., Gkelias, A. et Leung, K. K. (2017). A survey of indoor localization systems and technologies. *CoRR*, abs/1709.01015. Récupéré le 2021-05-10 de <http://arxiv.org/abs/1709.01015>

Zeiler, M. D. et Fergus, R. (2013). Visualizing and understanding convolutional networks. *CoRR*, *abs/1311.2901*. Récupéré le 2021-05-10 de <http://arxiv.org/abs/1311.2901>

APPENDICE A

MATRICES DE CONFUSION

		Activité Prédite																		
		Boire	Dormir	Enfiler une veste	Faire le ménage	Faire à manger	Faire du thé	Faire la vaisselle	Laver les dents	Laver les mains	Lire	Manger	Marcher	Mettre soulier	Prendre médicaments	Utiliser Ordinateur	Précision	Rappel	F1-Score	Nb. Instances
Activité Actuelle	Boire	0	0	0	0	0	0	0	0	0	30	8	0	0	0	22	0	0	0	60
	Dormir	0	14	0	0	0	0	134	0	38	0	0	0	0	0	0	0.67	0.08	0.14	186
	Enfiler une veste	0	0	0	0	10	19	25	0	0	0	0	0	12	0	0	0	0	0	66
	Faire le ménage	0	0	0	295	32	80	0	0	0	0	0	0	13	0	0	0.82	0.7	0.76	420
	Faire à manger	0	0	0	2	676	22	1	99	0	212	21	0	0	0	113	0.62	0.59	0.6	1146
	Faire du thé	0	1	0	5	217	51	0	166	0	90	97	0	2	0	31	0.2	0.08	0.11	660
	Faire la vaisselle	0	0	0	0	34	32	228	0	0	0	114	0	0	0	0	0.82	0.56	0.67	408
	Laver les dents	0	0	0	0	0	0	0	660	0	0	0	0	0	0	0	0.38	1	0.55	660
	Laver les mains	0	0	0	0	5	6	17	0	0	0	38	0	0	0	0	0	0	0	66
	Lire	0	6	0	0	17	0	0	266	0	107	0	0	0	0	6	0.21	0.27	0.23	402
	Manger	0	0	0	0	70	0	6	0	0	37	229	0	0	0	78	0.41	0.55	0.47	420
	Marcher	0	0	0	48	0	0	0	12	0	0	0	0	0	0	0	0	0	0	60
	Mettre soulier	0	0	0	9	30	47	0	0	0	0	0	0	34	0	0	0.56	0.28	0.38	120
	Prendre médicaments	0	0	0	0	0	0	0	0	0	4	54	0	0	0	2	0	0	0	60
	Utiliser Ordinateur	0	0	0	0	0	0	0	420	0	0	0	0	0	0	0	0	0	0	420

Figure A.1 : Matrice de confusion du modèle CNN-LSTM sur le jeu de données LOSO 1.

		Activité Prédite																		
		Laver les dents	Faire le ménage	Faire à manger	Faire la vaisselle	Boire	Manger	Faire du thé	Enfiler une veste	Mettre soulier	Lire	Dormir	Prendre Médicaments	Utiliser Ordinateurs	Marcher	Laver les mains	Précision	Retour	F1-Score	Nb. Instances
Activité Actuelle	Laver les dents	104	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.92	1	0.96	104
	Faire le ménage	0	82	0	0	0	3	0	0	0	6	0	3	0	11	0	0.93	0.78	0.85	105
	Faire à manger	0	0	86	0	0	0	12	0	0	0	0	0	4	0	0	0.7	0.84	0.76	102
	Faire la vaisselle	0	0	1	99	0	0	0	0	0	0	0	0	0	0	1	0.86	0.98	0.92	101
	Boire	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	15
	Manger	0	0	0	0	0	73	0	0	0	0	0	0	32	0	0	0.78	0.7	0.74	105
	Faire du thé	0	0	11	0	0	0	41	0	0	49	0	0	0	0	3	0.63	0.39	0.49	104
	Enfiler une veste	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	1	1	1	15
	Mettre soulier	0	6	0	0	0	0	0	0	23	0	0	0	0	0	0	0.62	0.79	0.7	29
	Lire	0	0	5	0	0	2	6	0	14	68	0	0	5	0	0	0.49	0.68	0.57	100
	Dormir	9	0	2	0	0	0	0	0	0	0	34	0	0	0	0	0.64	0.76	0.69	45
	Prendre Médicaments	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	15
Utiliser Ordinateurs	0	0	18	0	0	0	6	0	0	15	19	0	47	0	0	0.46	0.45	0.45	105	
Marcher	0	0	0	0	0	0	0	0	0	0	0	0	15	0	0	0.58	1	0.73	15	
Laver les mains	0	0	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	
		Activité Prédite																		

Figure A.2 : Matrice de confusion du modèle Tuned EfficientNetB0 avec LSTM sur le jeu de données LOSO 1.

APPENDICE B

TAUX DE RECONNAISSANCE

	LOSO 1			LOSO 2			LOSO 3		
Instance	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.52	0.80	0.90	0.47	0.67	0.67	0.57	0.87	0.93
2	0.65	0.82	0.93	0.48	0.55	0.72	0.58	0.80	0.95
3	0.42	0.67	0.83	0.43	0.67	0.93	0.55	0.83	0.98
4	0.52	0.75	0.95	0.52	0.63	0.77	0.63	0.82	0.95
5	0.47	0.68	0.83	0.47	0.62	0.77	0.65	0.90	1.00
Moyenne	0.51	0.74	0.89	0.47	0.63	0.77	0.60	0.84	0.96
Écart Type	0.08	0.06	0.05	0.03	0.04	0.09	0.04	0.04	0.02

	LOSO 4			LOSO 5			LOSO 6		
Instance	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.45	0.58	0.80	0.65	0.80	0.92	0.63	0.90	0.98
2	0.58	0.72	0.90	0.67	0.88	0.93	0.63	0.80	0.92
3	0.48	0.83	0.92	0.73	0.95	0.98	0.65	0.78	0.87
4	0.43	0.63	0.80	0.68	0.82	0.88	0.67	0.87	0.92
5	0.57	0.80	0.87	0.63	0.90	0.97	0.67	0.97	1.00
Moyenne	0.50	0.71	0.86	0.67	0.87	0.94	0.65	0.86	0.94
Écart Type	0.06	0.10	0.05	0.03	0.06	0.04	0.01	0.07	0.05

	LOSO 7			LOSO 8			LOSO 9		
Instance	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.65	0.97	1.00	0.35	0.62	0.73	0.55	0.88	0.98
2	0.63	0.90	1.00	0.42	0.63	0.80	0.57	0.93	0.97
3	0.55	0.93	0.95	0.50	0.75	1.00	0.63	0.82	0.93
4	0.67	0.97	1.00	0.55	0.70	0.80	0.58	0.97	1.00
5	0.62	1.00	1.00	0.42	0.72	0.87	0.55	0.82	0.98
Moyenne	0.62	0.95	0.99	0.45	0.68	0.84	0.58	0.88	0.97
Écart Type	0.04	0.03	0.02	0.07	0.05	0.09	0.03	0.06	0.02

	LOSO 10			LOSO 11			LOSO 12		
Instance	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.55	0.70	0.87	0.57	0.80	0.90	0.53	0.85	0.93
2	0.47	0.73	0.85	0.70	0.87	0.95	0.55	0.85	0.93
3	0.52	0.70	0.90	0.55	0.77	0.90	0.63	0.88	0.95
4	0.53	0.75	0.92	0.53	0.90	1.00	0.57	0.73	1.00
5	0.53	0.75	0.88	0.57	0.90	0.97	0.60	0.82	0.88
Moyenne	0.52	0.73	0.88	0.58	0.85	0.94	0.58	0.83	0.94
Écart Type	0.03	0.02	0.02	0.06	0.05	0.04	0.04	0.05	0.04

Figure B.1 : Taux de reconnaissance du modèle Tuned EfficientNetB0 sur le jeu de données de mouvements avec radars UWB installés à 36 cm du sol.

	LOSO 1			LOSO 2			LOSO 3		
Instance	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.55	0.57	0.63	0.62	0.88	1.00	0.73	0.80	0.93
2	0.68	0.80	0.88	0.68	0.90	1.00	0.78	0.85	1.00
3	0.67	1.00	1.00	0.63	0.73	0.93	0.87	0.95	1.00
4	0.82	0.97	1.00	0.58	0.73	0.83	0.73	0.75	0.75
5	0.70	0.75	1.00	0.68	0.77	0.78	0.70	0.82	0.85
Moyenne	0.68	0.82	0.90	0.64	0.80	0.91	0.76	0.83	0.91
Écart Type	0.08	0.16	0.14	0.04	0.07	0.09	0.06	0.07	0.10

	LOSO 4			LOSO 5			LOSO 6		
Instance	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.63	0.72	0.88	0.68	0.87	1.00	0.80	1.00	1.00
2	0.72	0.82	0.90	0.65	0.77	1.00	0.88	0.98	1.00
3	0.65	0.87	1.00	0.80	0.92	1.00	0.92	1.00	1.00
4	0.63	0.68	0.77	0.70	0.92	1.00	0.78	1.00	1.00
5	0.55	0.70	0.80	0.67	0.80	0.97	0.82	0.95	1.00
Moyenne	0.64	0.76	0.87	0.70	0.85	0.99	0.84	0.99	1.00
Écart Type	0.05	0.07	0.08	0.05	0.06	0.01	0.05	0.02	0.00

	LOSO 7			LOSO 8			LOSO 9		
Instance	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.88	0.98	1.00	0.58	0.73	0.98	0.90	1.00	1.00
2	0.88	0.98	1.00	0.42	0.68	1.00	0.82	1.00	1.00
3	0.95	1.00	1.00	0.43	0.62	0.88	0.90	1.00	1.00
4	0.90	1.00	1.00	0.43	0.77	0.87	0.80	1.00	1.00
5	0.88	1.00	1.00	0.43	0.77	0.88	0.90	1.00	1.00
Moyenne	0.90	0.99	1.00	0.46	0.71	0.92	0.86	1.00	1.00
Écart Type	0.03	0.01	0.00	0.06	0.06	0.06	0.05	0.00	0.00

	LOSO 10			LOSO 11			LOSO 12		
Instance	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.62	0.97	1.00	0.77	0.95	0.98	0.80	1.00	1.00
2	0.63	0.88	0.93	0.80	1.00	1.00	0.93	1.00	1.00
3	0.70	0.98	0.98	0.77	1.00	1.00	0.75	0.98	1.00
4	0.68	0.90	1.00	0.85	0.98	1.00	0.87	1.00	1.00
5	0.68	0.93	1.00	0.75	0.97	1.00	0.82	0.98	1.00
Moyenne	0.66	0.93	0.98	0.79	0.98	1.00	0.83	0.99	1.00
Écart Type	0.03	0.04	0.03	0.04	0.02	0.01	0.06	0.01	0.00

Figure B.2 : Taux de reconnaissance du modèle Tuned EfficientNetB0 avec entraînement par transfert sur le jeu de données de mouvements avec radars UWB installés à 36 cm du sol.

Instance	LOSO 1			LOSO 2			LOSO 3		
	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.67	0.88	1.00	0.63	0.88	0.95	0.67	0.87	0.98
2	0.53	0.88	0.93	0.58	0.77	0.92	0.67	0.87	0.93
3	0.55	0.77	0.90	0.55	0.83	0.97	0.73	0.92	0.95
4	0.60	0.97	0.98	0.53	0.68	0.90	0.58	0.72	0.90
5	0.60	0.85	0.93	0.65	0.85	0.95	0.65	0.80	0.90
Moyenne	0.59	0.87	0.95	0.59	0.80	0.94	0.66	0.83	0.93
Écart Type	0.05	0.06	0.04	0.05	0.07	0.02	0.05	0.07	0.03

Instance	LOSO 4			LOSO 5			LOSO 6		
	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.50	0.82	0.95	0.68	0.92	0.93	0.65	0.80	0.95
2	0.55	0.65	0.90	0.58	0.68	0.87	0.62	0.87	1.00
3	0.60	0.88	0.95	0.60	0.75	0.93	0.57	0.80	0.93
4	0.62	0.93	0.98	0.62	0.62	0.85	0.68	0.97	0.97
5	0.58	0.73	0.88	0.60	0.88	0.97	0.72	0.95	1.00
Moyenne	0.57	0.80	0.93	0.62	0.77	0.91	0.65	0.88	0.97
Écart Type	0.04	0.10	0.04	0.03	0.11	0.04	0.05	0.07	0.03

Instance	LOSO 7			LOSO 8			LOSO 9		
	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.63	0.88	0.97	0.40	0.80	0.98	0.75	0.88	0.98
2	0.67	0.93	0.97	0.50	0.67	0.88	0.57	0.82	0.93
3	0.58	0.85	0.95	0.38	0.52	0.77	0.68	0.93	0.97
4	0.48	0.75	0.88	0.45	0.70	0.83	0.62	0.92	0.98
5	0.65	0.90	0.97	0.48	0.72	0.87	0.72	0.90	0.97
Moyenne	0.60	0.86	0.95	0.44	0.68	0.87	0.67	0.89	0.97
Écart Type	0.07	0.06	0.03	0.05	0.09	0.07	0.07	0.04	0.02

Instance	LOSO 10			LOSO 11			LOSO 12		
	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.67	0.83	0.95	0.70	0.88	0.98	0.58	0.78	0.92
2	0.68	0.92	0.97	0.62	0.82	0.98	0.57	0.83	0.93
3	0.63	0.92	0.98	0.73	0.98	1.00	0.52	0.78	0.98
4	0.73	0.95	0.97	0.67	0.82	0.95	0.43	0.63	0.77
5	0.63	0.97	1.00	0.75	0.98	1.00	0.47	0.70	0.95
Moyenne	0.67	0.92	0.97	0.69	0.90	0.98	0.51	0.75	0.91
Écart Type	0.04	0.05	0.02	0.05	0.07	0.02	0.06	0.07	0.08

Figure B.3 : Taux de reconnaissance du modèle Tuned EfficientNetB0 sur le jeu de données de mouvements avec radars UWB installés à 96 cm du sol.

Instance	LOSO 1			LOSO 2			LOSO 3		
	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.85	1.00	1.00	0.78	0.85	1.00	0.77	0.97	1.00
2	0.87	0.93	1.00	0.80	0.92	1.00	0.73	0.75	0.75
3	0.92	0.97	0.98	0.87	0.95	1.00	0.75	0.80	0.88
4	0.73	0.88	1.00	0.75	0.75	0.77	0.65	0.73	0.82
5	0.65	0.75	0.83	0.75	0.75	0.75	0.82	0.85	1.00
Moyenne	0.80	0.91	0.96	0.79	0.84	0.90	0.74	0.82	0.89
Écart Type	0.10	0.09	0.07	0.04	0.08	0.12	0.05	0.08	0.10

Instance	LOSO 4			LOSO 5			LOSO 6		
	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.72	0.98	1.00	0.73	0.83	0.97	0.70	0.92	0.93
2	0.80	0.97	1.00	0.68	0.75	0.85	0.72	0.78	0.90
3	0.63	0.95	1.00	0.85	0.97	0.98	0.75	0.98	1.00
4	0.78	0.98	1.00	0.72	0.98	0.98	0.83	0.97	1.00
5	0.65	0.88	1.00	0.67	0.75	0.75	0.78	0.97	1.00
Moyenne	0.72	0.95	1.00	0.73	0.86	0.91	0.76	0.92	0.97
Écart Type	0.07	0.04	0.00	0.06	0.10	0.09	0.05	0.07	0.04

Instance	LOSO 7			LOSO 8			LOSO 9		
	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.95	1.00	1.00	0.47	0.55	0.75	0.88	0.97	1.00
2	0.93	1.00	1.00	0.50	0.68	0.75	0.85	0.98	1.00
3	0.90	1.00	1.00	0.50	0.72	0.85	0.87	0.95	1.00
4	0.77	1.00	1.00	0.27	0.42	0.63	0.87	1.00	1.00
5	0.85	1.00	1.00	0.52	0.88	0.97	0.88	1.00	1.00
Moyenne	0.88	1.00	1.00	0.45	0.65	0.79	0.87	0.98	1.00
Écart Type	0.07	0.00	0.00	0.09	0.16	0.11	0.01	0.02	0.00

Instance	LOSO 10			LOSO 11			LOSO 12		
	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3	Top-1	Top-2	Top-3
1	0.75	0.85	1.00	0.88	0.97	1.00	0.67	0.93	1.00
2	0.67	1.00	1.00	0.88	1.00	1.00	0.65	0.77	0.88
3	0.73	0.93	1.00	0.95	1.00	1.00	0.68	0.95	0.98
4	0.68	0.95	1.00	0.98	1.00	1.00	0.67	0.97	0.98
5	0.65	0.78	1.00	0.88	1.00	1.00	0.67	0.85	0.97
Moyenne	0.70	0.90	1.00	0.92	0.99	1.00	0.67	0.89	0.96
Écart Type	0.04	0.08	0.00	0.04	0.01	0.00	0.01	0.07	0.04

Figure B.4 : Taux de reconnaissance du modèle Tuned EfficientNetB0 avec entraînement par transfert sur le jeu de données de mouvements avec radars UWB installés à 96 cm du sol.

APPENDICE C

MATRICES DE CONFUSION SUR LE JEU DE DONNÉES DE MOUVEMENTS

		Classe Prédite																								
		LOSO 1				LOSO 2				LOSO 3				LOSO 4				LOSO 5				LOSO 6				
		0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	
Classe Actuelle	Instance 1	0	8	7	0	0	14	0	0	1	7	8	0	0	15	0	0	0	11	2	2	0	14	1	0	0
		1	2	8	1	4	4	5	2	4	0	14	1	0	4	5	1	5	6	6	2	1	3	5	4	3
		2	3	1	4	7	5	0	4	6	0	6	9	0	6	2	0	7	0	2	11	2	0	0	7	8
		3	0	3	1	11	7	0	3	5	0	4	7	4	4	4	0	7	0	0	4	11	0	0	3	12
	Instance 2	0	12	3	0	0	15	0	0	0	2	11	0	2	15	0	0	0	13	2	0	0	7	6	2	0
		1	1	12	2	0	2	13	0	0	5	10	0	0	13	2	0	0	6	4	3	2	0	12	2	1
		2	1	3	10	1	7	5	1	2	0	0	12	3	1	1	4	9	0	1	11	3	0	4	7	4
		3	1	5	4	5	9	5	1	0	0	1	3	11	0	0	1	14	0	0	3	12	0	1	2	12
	Instance 3	0	4	9	0	2	11	2	1	1	11	4	0	0	4	9	1	1	15	0	0	0	11	3	1	0
		1	3	12	0	0	0	2	13	0	3	12	0	0	1	10	3	1	7	6	1	1	1	14	0	0
		2	2	10	2	1	0	1	12	2	0	5	0	10	0	4	9	2	0	2	12	1	1	2	5	7
		3	2	5	1	7	0	3	11	1	0	3	2	10	0	2	7	6	0	0	4	11	3	0	3	9
	Instance 4	0	10	3	0	2	13	2	0	0	13	2	0	0	14	1	0	0	13	2	0	0	7	7	1	0
		1	0	12	2	1	0	14	1	0	3	12	0	0	5	6	3	1	3	3	5	4	3	8	4	0
		2	3	7	5	0	0	11	3	1	0	2	6	7	0	9	6	0	1	0	11	3	0	0	11	4
		3	1	3	7	4	0	12	2	1	0	4	4	7	1	0	14	0	0	0	1	14	0	0	1	14
	Instance 5	0	11	3	0	1	15	0	0	0	9	6	0	0	15	0	0	0	13	2	0	0	10	5	0	0
		1	3	7	2	3	2	7	3	3	1	10	4	0	8	7	0	0	7	7	1	0	2	10	3	0
		2	1	8	3	3	5	1	5	4	0	4	10	1	5	2	7	1	0	0	8	7	0	0	8	7
		3	0	1	7	7	8	3	3	1	0	0	5	10	2	5	3	5	0	0	5	10	0	0	3	12

Figure C.1 : Matrices de confusion du modèle Tuned EfficientNetB0 sur le jeu de données de mouvements avec radars UWB installés à 36 cm du sol, LOSO 1 à 6.

		Classe Prédite																								
		LOSO 7				LOSO 8				LOSO 9				LOSO 10				LOSO 11				LOSO 12				
		0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	
Classe Actuelle	Instance 1	0	8	7	0	0	11	2	2	0	9	5	1	0	12	3	0	0	10	2	1	2	5	8	2	0
		1	4	11	0	0	14	1	0	0	6	8	1	0	3	8	0	4	7	8	0	0	0	11	3	1
		2	0	3	6	6	6	0	9	0	0	0	5	10	2	4	3	6	3	0	9	3	2	4	7	2
		3	0	0	1	14	10	1	4	0	0	0	4	11	2	1	2	10	0	2	6	7	0	0	6	9
	Instance 2	0	13	2	0	0	3	1	0	11	7	8	0	0	0	10	5	0	9	3	2	1	5	7	2	1
		1	5	8	1	1	0	8	0	7	4	9	0	2	0	8	6	1	2	12	0	1	0	12	3	0
		2	0	1	4	10	0	6	0	9	0	0	5	10	0	1	12	2	0	0	12	3	0	6	4	5
		3	0	0	2	13	0	1	0	14	0	0	2	13	0	0	7	8	0	0	6	9	0	1	2	12
	Instance 3	0	11	4	0	0	6	3	5	1	9	6	0	0	11	0	2	2	13	2	0	0	11	2	1	1
		1	14	1	0	0	3	7	4	1	6	9	0	0	2	3	4	6	5	6	3	1	2	12	1	0
		2	1	0	8	6	2	0	10	3	0	1	6	8	0	1	10	4	1	1	6	7	3	5	7	0
		3	0	0	2	13	1	0	7	7	0	0	1	14	1	1	6	7	0	0	7	8	1	0	6	8
	Instance 4	0	11	3	1	0	9	3	2	1	10	5	0	0	14	1	0	0	14	0	0	1	11	4	0	0
		1	7	8	0	0	2	8	4	1	8	7	0	0	7	2	4	2	9	4	1	1	4	10	0	1
		2	0	0	6	9	1	1	12	1	0	0	9	6	1	3	7	4	0	0	8	7	6	9	0	0
		3	0	0	0	15	1	6	4	4	0	1	5	9	0	1	5	9	2	0	7	6	0	2	0	13
	Instance 5	0	10	5	0	0	11	3	1	0	13	0	1	1	13	2	0	0	11	0	0	4	5	10	0	0
		1	2	13	0	0	8	7	0	0	5	2	6	2	4	11	0	0	5	7	1	2	0	15	0	0
		2	0	0	7	8	4	4	7	0	0	0	3	12	5	4	3	3	1	0	4	10	1	3	6	5
		3	0	0	8	7	2	2	11	0	0	0	0	15	5	5	0	5	0	0	3	12	0	0	5	10

Figure C.2 : Matrices de confusion du modèle Tuned EfficientNetB0 sur le jeu de données de mouvements avec radars UWB installés à 36 cm du sol, LOSO 7 à 12.

		Classe Prédite																									
		LOSO 1				LOSO 2				LOSO 3				LOSO 4				LOSO 5				LOSO 6					
		0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3		
Classe Actuelle	Instance 1	0	12	3	0	0	9	3	3	0	12	3	0	0	13	2	0	0	15	0	0	0	15	0	0	0	
		1	1	13	1	0	0	12	2	1	0	12	3	0	11	2	1	1	3	11	1	0	6	9	0	0	
		2	0	6	9	0	0	0	11	4	2	1	8	4	0	1	9	5	1	2	9	3	2	0	13	0	
		3	0	2	7	6	0	2	7	6	1	1	5	8	0	0	9	6	0	0	9	6	3	4	6	2	
	Instance 2	0	11	4	0	0	14	1	0	0	15	0	0	0	8	7	0	0	11	3	1	0	12	3	0	0	
		1	4	9	2	0	6	7	2	0	3	10	1	1	1	14	0	0	1	14	0	0	7	8	0	0	
		2	2	5	8	0	1	3	8	3	4	2	8	1	0	8	0	7	0	5	10	0	0	0	15	0	
		3	2	3	6	4	1	5	3	6	0	5	3	7	0	2	2	11	0	7	8	0	0	0	13	2	
	Instance 3	0	13	1	1	0	15	0	0	0	13	2	0	0	5	8	0	2	11	3	1	0	14	1	0	0	
		1	7	4	3	1	11	3	0	1	2	8	1	4	0	14	1	0	1	13	1	0	11	2	2	0	
		2	4	2	6	3	1	0	9	5	1	1	13	0	0	5	7	3	2	4	4	5	0	1	11	3	
		3	0	0	5	10	0	0	9	6	0	0	5	10	0	1	4	10	1	3	3	8	0	3	5	7	
	Instance 4	0	11	0	0	4	15	0	0	0	13	1	0	1	7	6	2	0	13	2	0	0	12	1	2	0	
		1	5	10	0	0	6	2	5	2	3	7	4	1	2	13	0	0	7	4	2	2	6	9	0	0	
		2	0	4	4	7	0	3	9	3	2	2	5	6	0	8	3	4	0	0	12	3	0	0	15	0	
		3	0	1	3	11	1	4	4	6	1	1	3	10	0	0	1	14	0	1	6	8	0	0	10	5	
	Instance 5	0	11	4	0	0	14	1	0	0	14	1	0	0	12	3	0	0	8	7	0	0	11	4	0	0	
		1	4	7	4	0	1	13	0	1	7	5	1	2	5	10	0	0	0	12	2	1	0	13	2	0	
		2	1	5	9	0	0	5	7	3	5	1	7	2	0	4	4	7	0	5	4	6	0	0	10	5	
		3	0	0	6	9	0	6	4	5	0	0	2	13	4	1	1	9	1	1	1	12	0	0	6	9	

Figure C.3 : Matrices de confusion du modèle Tuned EfficientNetB0 sur le jeu de données de mouvements avec radars UWB installés à 96 cm du sol, LOSO 1 à 6.

		Classe Prédite																											
		LOSO 7				LOSO 8				LOSO 9				LOSO 10				LOSO 11				LOSO 12							
		0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3				
Classe Actuelle	Instance 1	0	10	4	1	0	1	14	0	0	11	2	1	1	10	3	2	0	9	2	4	0	7	5	2	1			
		1	0	8	3	4	6	7	2	0	2	10	3	0	6	9	0	0	1	10	4	0	3	5	5	2			
		2	0	0	8	7	0	2	13	0	1	0	10	4	0	0	6	9	0	0	9	6	0	1	11	3			
		3	1	0	2	12	0	4	8	3	0	0	1	14	0	0	0	15	0	0	1	14	1	0	2	12			
	Instance 2	0	12	3	0	0	14	1	0	0	4	7	1	3	6	4	4	1	12	3	0	0	14	1	0	0			
		1	1	10	3	1	6	0	8	1	0	11	1	3	1	11	1	2	4	11	0	0	7	7	1	0			
		2	0	2	3	10	3	1	4	7	0	3	7	5	0	0	13	2	0	3	6	6	6	0	4	5			
		3	0	0	0	15	1	0	2	12	0	1	2	12	0	0	4	11	0	0	7	8	1	2	3	9			
	Instance 3	0	11	4	0	0	13	2	0	0	12	3	0	0	9	5	0	1	11	4	0	0	12	3	0	0			
		1	6	9	0	0	7	7	0	1	2	11	1	1	4	10	0	1	5	10	0	0	11	4	0	0			
		2	0	2	4	9	6	6	3	0	0	1	5	9	0	0	7	8	0	4	11	0	5	1	9	0			
		3	0	4	0	11	4	11	0	0	0	0	2	13	0	0	3	12	0	0	3	12	3	0	6	6			
	Instance 4	0	8	3	0	4	12	3	0	0	8	6	1	0	11	4	0	0	13	2	0	0	0	2	8	5			
		1	4	8	0	3	5	9	0	1	2	9	4	0	4	11	0	0	3	12	0	0	0	3	6	6			
		2	3	1	4	7	3	5	5	2	1	5	9	0	0	0	7	8	1	6	8	0	0	0	12	3			
		3	4	2	0	9	6	4	4	1	0	0	4	11	0	0	0	15	2	3	3	7	0	0	4	11			
	Instance 5	0	11	4	0	0	10	5	0	0	8	6	1	0	1	13	1	0	14	1	0	0	12	1	2	0			
		1	3	9	3	0	3	9	1	2	1	9	4	1	1	14	0	0	4	10	1	0	6	7	1	1			
		2	0	0	10	5	11	1	1	2	0	0	13	2	0	0	10	5	0	1	10	4	11	0	4	0			
		3	0	1	5	9	0	0	6	9	0	0	2	13	0	0	2	13	0	0	4	11	5	0	5	5			

Figure C.4 : Matrices de confusion du modèle Tuned EfficientNetB0 sur le jeu de données de mouvements avec radars UWB installés à 96 cm du sol, LOSO 7 à 12.

		Classe Prédite																								
		LOSO 1				LOSO 2				LOSO 3				LOSO 4				LOSO 5				LOSO 6				
		0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	
Classe Actuelle	Instance 1	0	15	0	0	0	15	0	0	0	14	1	0	0	15	0	0	0	12	3	0	0	11	4	0	0
		1	1	3	1	10	0	4	10	1	1	14	0	0	14	0	1	0	0	12	0	3	0	15	0	0
		2	0	3	0	12	0	0	6	9	0	6	1	8	7	0	8	0	0	1	10	4	0	0	15	0
		3	0	0	0	15	0	0	3	12	0	0	0	15	0	0	0	15	0	3	5	7	0	0	8	7
	Instance 2	0	15	0	0	0	12	0	0	3	15	0	0	0	15	0	0	0	15	0	0	0	15	0	0	0
		1	1	13	1	0	0	14	1	0	0	15	0	0	13	0	1	1	0	0	11	4	1	10	4	0
		2	0	2	13	0	0	0	0	15	0	12	2	1	0	0	13	2	0	0	12	3	0	0	13	2
		3	0	0	15	0	0	0	0	15	0	0	0	15	0	0	0	15	0	0	3	12	0	0	0	15
	Instance 3	0	15	0	0	0	14	1	0	0	15	0	0	0	10	0	5	0	14	1	0	0	15	0	0	0
		1	0	12	3	0	0	13	0	2	4	11	0	0	7	0	8	0	1	11	0	3	0	13	2	0
		2	0	2	13	0	0	1	0	14	0	0	11	4	0	1	14	0	0	3	9	3	0	1	13	1
		3	0	0	15	0	0	4	0	11	0	0	0	15	0	0	0	15	0	0	1	14	0	0	1	14
	Instance 4	0	15	0	0	0	9	6	0	0	14	1	0	0	15	0	0	0	15	0	0	0	15	0	0	0
		1	2	10	3	0	0	11	3	1	0	15	0	0	7	0	8	0	3	11	1	0	1	13	1	0
		2	0	2	13	0	1	6	0	8	0	0	0	15	0	3	8	4	0	0	15	0	0	0	9	6
		3	0	0	4	11	0	0	0	15	0	0	0	15	0	0	0	15	0	2	12	1	0	0	5	10
	Instance 5	0	15	0	0	0	15	0	0	0	14	1	0	0	15	0	0	0	15	0	0	0	12	3	0	0
		1	0	13	2	0	0	15	0	0	2	13	0	0	15	0	0	0	0	15	0	0	0	15	0	0
		2	0	1	14	0	3	3	1	8	0	0	0	15	9	1	3	2	0	4	0	11	0	1	9	5
		3	0	0	15	0	4	0	1	10	0	0	0	15	0	0	0	15	0	5	0	10	0	2	0	13

Figure C.5 : Matrices de confusion du modèle Tuned EfficientNetB0 avec apprentissage par transfert sur le jeu de données de mouvements avec radars UWB installés à 36 cm du sol, LOSO 1 à 6.

		Classe Prédite																								
		LOSO 7				LOSO 8				LOSO 9				LOSO 10				LOSO 11				LOSO 12				
		0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	
Classe Actuelle	Instance 1	0	15	0	0	0	15	0	0	0	15	0	0	0	3	12	0	0	15	0	0	0	12	3	0	0
		1	0	15	0	0	15	0	0	0	0	15	0	0	1	12	2	0	0	15	0	0	0	15	0	0
		2	0	0	12	3	0	1	12	2	0	0	13	2	0	1	11	3	0	0	2	13	0	1	9	5
		3	0	0	4	11	0	0	7	8	0	0	4	11	0	0	4	11	0	0	1	14	0	0	3	12
	Instance 2	0	15	0	0	0	9	6	0	0	15	0	0	0	14	1	0	0	14	1	0	0	15	0	0	0
		1	0	15	0	0	11	4	0	0	0	15	0	0	5	3	1	6	0	15	0	0	2	12	1	0
		2	0	1	9	5	0	3	12	0	0	0	5	10	0	0	12	3	0	0	4	11	0	0	14	1
		3	0	0	1	14	0	7	8	0	0	0	1	14	0	0	6	9	0	0	0	15	0	0	0	15
	Instance 3	0	15	0	0	0	13	1	1	0	15	0	0	0	12	3	0	0	14	1	0	0	15	0	0	0
		1	0	15	0	0	15	0	0	0	2	13	0	0	9	4	2	0	0	15	0	0	12	3	0	0
		2	0	0	15	0	2	0	13	0	0	0	15	0	0	0	15	0	0	0	2	13	1	0	13	1
		3	0	0	3	12	0	0	15	0	0	0	4	11	0	0	4	11	0	0	0	15	0	0	1	14
	Instance 4	0	15	0	0	0	4	11	0	0	15	0	0	0	10	5	0	0	12	3	0	0	15	0	0	0
		1	0	15	0	0	2	13	0	0	0	15	0	0	1	7	2	5	0	15	0	0	8	7	0	0
		2	0	0	10	5	0	7	8	0	0	0	15	0	0	0	10	5	0	0	9	6	0	0	15	0
		3	0	0	1	14	0	4	10	1	0	0	12	3	0	0	1	14	0	0	0	15	0	0	0	15
	Instance 5	0	15	0	0	0	15	0	0	0	15	0	0	0	8	5	2	0	15	0	0	0	12	3	0	0
		1	0	13	2	0	15	0	0	0	1	14	0	0	3	12	0	0	0	15	0	0	0	15	0	0
		2	0	0	11	4	8	5	1	1	0	1	13	1	0	0	6	9	0	0	0	15	0	2	13	0
		3	0	0	1	14	0	5	0	10	0	0	3	12	0	0	0	15	0	0	0	15	0	0	6	9

Figure C.6 : Matrices de confusion du modèle Tuned EfficientNetB0 avec apprentissage par transfert sur le jeu de données de mouvements avec radars UWB installés à 36 cm du sol, LOSO 7 à 12.

		Classe Prédite																								
		LOSO 1				LOSO 2				LOSO 3				LOSO 4				LOSO 5				LOSO 6				
		0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	
Classe Actuelle	Instance 1	0	13	0	2	0	15	0	0	0	15	0	0	0	12	3	0	0	15	0	0	0	13	2	0	0
		1	0	14	1	0	3	2	10	0	0	11	1	3	1	14	0	0	13	0	2	0	0	14	0	1
		2	0	2	13	0	0	0	15	0	0	0	14	1	0	9	6	0	1	0	14	0	0	0	0	15
		3	0	0	4	11	0	0	0	15	0	1	8	6	0	0	4	11	0	0	0	15	0	0	0	15
	Instance 2	0	12	3	0	0	15	0	0	0	15	0	0	0	13	1	0	1	14	1	0	0	14	1	0	0
		1	0	15	0	0	6	6	3	0	0	14	0	1	4	11	0	0	0	15	0	0	0	14	1	0
		2	0	4	10	1	0	0	12	3	0	0	0	15	0	1	11	3	1	2	12	0	0	0	0	15
		3	0	0	0	15	0	0	0	15	0	0	0	15	0	1	1	13	0	3	12	0	0	0	0	15
	Instance 3	0	15	0	0	0	15	0	0	0	15	0	0	0	5	10	0	0	15	0	0	0	10	2	0	3
		1	0	14	1	0	2	9	4	0	0	15	0	0	2	12	1	0	0	14	1	0	1	11	1	2
		2	0	2	11	2	0	0	14	1	1	9	0	5	0	2	13	0	0	1	14	0	0	0	9	6
		3	0	0	0	15	0	0	1	14	0	0	0	15	0	0	7	8	2	1	4	8	0	0	0	15
	Instance 4	0	15	0	0	0	15	0	0	0	15	0	0	0	15	0	0	0	12	3	0	0	15	0	0	0
		1	0	9	4	2	15	0	0	0	1	10	0	4	13	2	0	0	0	15	0	0	4	11	0	0
		2	0	3	6	6	0	0	15	0	0	1	0	14	0	0	15	0	0	1	11	3	0	0	14	1
		3	0	1	0	14	0	0	0	15	1	0	0	14	0	0	0	15	10	0	0	5	0	0	5	10
	Instance 5	0	15	0	0	0	15	0	0	0	15	0	0	0	15	0	0	0	11	4	0	0	15	0	0	0
		1	0	9	0	6	5	0	8	2	1	14	0	0	6	9	0	0	0	15	0	0	10	5	0	0
		2	0	5	0	10	0	0	15	0	0	7	5	3	0	0	15	0	1	0	14	0	0	0	13	2
		3	0	0	0	15	0	0	0	15	0	0	0	15	0	0	15	0	2	1	12	0	0	0	1	14

Figure C.7 : Matrices de confusion du modèle Tuned EfficientNetB0 avec apprentissage par transfert sur le jeu de données de mouvements avec radars UWB installés à 96 cm du sol, LOSO 1 à 6.

		Classe Prédite																								
		LOSO 7				LOSO 8				LOSO 9				LOSO 10				LOSO 11				LOSO 12				
		0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3	
Classe Actuelle	Instance 1	0	15	0	0	0	14	0	1	0	10	5	0	0	8	5	0	2	15	0	0	0	11	3	1	0
		1	0	15	0	0	6	0	9	0	0	14	1	0	0	15	0	0	0	15	0	0	3	11	1	0
		2	0	1	14	0	1	0	14	0	0	0	14	1	0	0	7	8	0	3	12	0	0	0	15	0
		3	0	0	2	13	6	5	4	0	0	0	0	15	0	0	0	15	0	0	4	11	0	0	12	3
	Instance 2	0	15	0	0	0	15	0	0	0	13	2	0	0	11	4	0	0	15	0	0	0	12	3	0	0
		1	0	15	0	0	13	1	1	0	2	13	0	0	2	13	0	0	0	15	0	0	3	12	0	0
		2	0	0	14	1	0	0	15	0	0	0	10	5	0	0	1	14	0	0	14	1	0	0	0	15
		3	0	0	3	12	0	0	15	0	0	0	0	15	0	0	0	15	0	0	6	9	0	0	0	15
	Instance 3	0	15	0	0	0	15	0	0	0	15	0	0	0	14	1	0	0	15	0	0	0	13	2	0	0
		1	0	15	0	0	15	0	0	0	5	9	1	0	0	14	1	0	0	15	0	0	2	13	0	0
		2	0	0	13	2	15	0	0	0	0	0	13	2	0	0	1	14	0	0	15	0	0	0	15	0
		3	0	0	4	11	0	0	0	15	0	0	0	15	0	0	0	15	0	0	3	12	0	0	15	0
	Instance 4	0	15	0	0	0	15	0	0	0	14	1	0	0	15	0	0	0	15	0	0	0	14	1	0	0
		1	0	14	1	0	15	0	0	0	2	10	3	0	7	8	0	0	0	14	1	0	6	9	0	0
		2	0	0	6	9	15	0	0	0	0	0	14	1	0	0	6	9	0	0	15	0	0	0	5	10
		3	0	3	1	11	14	0	0	1	0	0	1	14	0	0	3	12	0	0	0	15	0	0	3	12
	Instance 5	0	13	2	0	0	15	0	0	0	14	1	0	0	13	2	0	0	13	0	0	2	15	0	0	0
		1	0	15	0	0	13	1	1	0	3	12	0	0	4	11	0	0	0	15	0	0	15	0	0	0
		2	0	0	15	0	0	0	15	0	0	0	12	3	0	0	0	15	0	0	15	0	1	0	10	4
		3	0	0	7	8	0	0	15	0	0	0	0	15	0	0	0	15	0	0	5	10	0	0	0	15

Figure C.8 : Matrices de confusion du modèle Tuned EfficientNetB0 avec apprentissage par transfert sur le jeu de données de mouvements avec radars UWB installés à 96 cm du sol, LOSO 7 à 12.

APPENDICE D

CERTIFICAT D'APPROBATION ÉTHIQUE

Le 21 mai 2019

À l'attention de :

Camille Bertuolia, Étudiant
Maîtrise en informatique
Département d'informatique et de mathématique

Kevin Bouchard, Professeur
Département d'informatique et de mathématique
Sébastien Gaboury, Professeur
Département d'informatique et de mathématique
Julien Maitre, Chargé de cours
Département d'informatique et de mathématique

Objet : Approbation éthique de votre projet de recherche

Projet : 2019-220

Titre du projet de recherche : Reconnaissance d'activités dans un habitat intelligent

Source de financement : Programme de soutien à la valorisation et au transfert (PSVT) du ministère de l'Économie et de l'Innovation (MESI) (financement de Kevin Bouchard)

Bonjour,

Votre projet de recherche a fait l'objet d'une évaluation en matière d'éthique de la recherche avec des êtres humains par le Comité d'éthique de la recherche de l'Université du Québec à Chicoutimi (CER-UQAC).

Un certificat d'approbation éthique qui atteste de la conformité de votre projet de recherche à la [Politique d'éthique de la recherche avec des êtres humains](#), de l'UQAC est émis en date du 21 mai 2019. Prenez note que ce certificat est **valide jusqu'au 21 mai 2020**.

Veuillez noter que le Formulaire d'information et de consentement a été revu et légèrement modifié par le CER-UQAC. Ce document sera déposé dans votre projet (voir les documents précédés d'un carré mauve dans la section "Fichiers" - "Documents officiels"). Cette version est celle autorisée par le CER et elle devra être utilisée pour votre projet.

Notez qu'en vertu de la [Politique d'éthique de la recherche avec des êtres humains](#), il est de la responsabilité des chercheurs d'assurer que leurs projets de recherche conservent une approbation éthique pour toute la durée des travaux de recherche et d'informer le CER de la fin de ceux-ci. Vous devrez donc obtenir le renouvellement de votre approbation éthique avant l'expiration de ce certificat à l'aide du formulaire *F7 - Renouvellement annuel*. Un rappel automatique vous sera envoyé par courriel quelques semaines avant l'échéance de votre certificat.

Un rapport non remis entraînera la fermeture du dossier éthique dans les 30 jours qui suivent et une mention y sera ajoutée. Ainsi, la poursuite de la **cueillette de données** auprès des participants, sans certification éthique valide, ou le fait d'**apporter une modification significative** (à la population ciblée, au formulaire de consentement, au protocole d'expérimentation, à la méthode de collecte ou de traitement des données, etc.) **ou affectant le niveau de risque du projet** sans approbation du CER-UQAC représentent des situations relevant

de la [Politique relative à la conduite responsable en recherche et en création](#). De plus, le chercheur a l'obligation de signaler tout incident grave dès qu'il survient.

Si des modifications sont apportées à votre projet avant l'échéance du certificat, vous devrez remplir le formulaire *F8 - Modification de projet* et obtenir l'approbation du CER avant de mettre en oeuvre ces modifications. Si votre projet est terminé avant l'échéance du certificat, vous devrez remplir le formulaire *F9 - Fin de projet*.

Enfin, veuillez prendre note que le Décanat des études est mis en copie conforme de ce courriel afin de l'aviser de l'obtention de votre certification éthique. De plus, puisque votre demande d'approbation indique l'obtention d'un financement, le Décanat de la recherche et de la création est mis en copie conforme afin de l'informer de l'obtention de votre certification éthique.

En vous souhaitant bon succès dans la réalisation de votre recherche, veuillez recevoir mes salutations distinguées.

Tommy Chevette,
Professeur et président du CER

CERTIFICAT D'APPROBATION ÉTHIQUE

La présente atteste que le projet de recherche décrit ci-dessous a fait l'objet d'une évaluation en matière d'éthique de la recherche avec des êtres humains et qu'il satisfait aux exigences de la politique de l'UQAC en cette matière.

Les membres jugent que ce projet rencontre les critères d'une recherche à risque minimal.

Projet # : 2019-220

Titre du projet de recherche : Reconnaissance d'activités dans un habitat intelligent

Chercheur principal :

Camille Bertuolia, Étudiant
Maîtrise en informatique
Département d'informatique et de mathématique

Directeur/codirecteur :

En provenance de l'UQAC:
Kevin Bouchard, Professeur
Département d'informatique et de mathématique
Sébastien Gaboury, Professeur
Département d'informatique et de mathématique
Julien Maitre, Chargé de cours
Département d'informatique et de mathématique

Financement : Programme de soutien à la valorisation et au transfert (PSVT) du ministère de l'Économie et de l'Innovation (MESI) (financement de Kevin Bouchard)

Date d'approbation du projet : 21 mai 2019

Date d'entrée en vigueur du certificat : 21 mai 2019

Date d'échéance du certificat : 21 mai 2020



Tommy Chevette